PLENOPTIC IMAGING AND VISION USING ANGLE SENSITIVE PIXELS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Suren Jayasuriya

January 2017

PLENOPTIC IMAGING AND VISION USING ANGLE SENSITIVE PIXELS

Suren Jayasuriya, Ph.D.

Cornell University 2017

Computational cameras with sensor hardware co-designed with computer vision and graphics algorithms are an exciting recent trend in visual computing. In particular, most of these new cameras capture the plenoptic function of light, a multidimensional function of radiance for light rays in a scene. Such plenoptic information can be used for a variety of tasks including depth estimation, novel view synthesis, and inferring physical properties of a scene that the light interacts with.

In this thesis, we present multimodal plenoptic imaging, the simultaenous capture of multiple plenoptic dimensions, using Angle Sensitive Pixels (ASP), custom CMOS image sensors with embedded per-pixel diffraction gratings. We extend ASP models for plenoptic image capture, and showcase several computer vision and computational imaging applications.

First, we show how high resolution 4D light fields can be recovered from ASP images, using both a dictionary-based machine learning method as well as deep learning. We then extend ASP imaging to include the effects of polarization, and use this new information to image stress-induced birefringence and remove specular highlights from light field depth mapping. We explore the potential for ASPs performing time-of-flight imaging, and introduce the depth field, a combined representation of time-of-flight depth with plenoptic spatio-angular coordinates, which is used for applications in robust depth estimation. Finally, we leverage ASP optical edge filtering to be a low power front end for an embedded deep learning imaging system. We also present two technical appendices: a study of using deep learning with energy-efficient binary gradient cameras, and a design flow to enable

agile hardware design for computational image sensors in the future.

BIOGRAPHICAL SKETCH

The author was born on July 19, 1990. From 2008 to 2012 he studied at the University of Pittsburgh, where he received a Bachelor of Science in Mathematics with departmental honors and a Bachelor of Arts in Philosophy. He then moved to Ithaca, New York to study at Cornell University from 2012 to 2017, where he earned his doctorate in Electrical and Computer Engineering in 2017.

To my parents and brother.

ACKNOWLEDGEMENTS

I am grateful for my advisor Al Molnar for all his mentoring over these past years. It was a risk to accept a student whose background was mathematics and philosophy, and had no experience in ECE or CS, but Al was patient enough to allow me to make mistakes and grow as a researcher. I also want to thank my commitee members Alyssa Apsel and Steve Marschner for their advice throughout my graduate career.

Several people made important contributions to this thesis. Sriram Sivaramakrishnan was my go-to expert on ASPs, and co-author on many of my imaging papers. Matthew Hirsch and Gordon Wetzstein, along with the supervision of Ramesh Raskar, collaborated on the dictionary-based learning for recovering 4D light fields in Chapter 3. Arjun Jauhari, Mayank Gupta, and Kuldeep Kulkarni, along with the supervision of Pavan Turaga, performed deep learning experiments to speed up light field reconstructions in Chapter 3. Ellen Chuang and Debashree Guruaribam performed the polarization experiments in Chapter 4. Adithya Pediredla and Ryuichi Tadano helped create the depth fields experimental prototype and worked on applications in Chapter 5. George Chen realized ASP Vision from conception with me, and Jiyue Yang and Judy Stephen collected and analyzed the real digit and face recognition experiments in Chapter 6. Ashok Veeraraghavan helped supervise the projects in Chapters 5 and 6.

The work in Appendix A was completed during a summer internship at NVIDIA Research under the direction of Orazio Gallo, Jinwei Gu, and Jan Kautz, with the face detection results compiled by Jinwei Gu, the gesture recognition results run by Pavlo Molchanov, and the captured gradient images for intensity reconstruction by Orazio Gallo. The design flow in Appendix B was developed in collaboration with Chris Torng, Moyang Wang, Nagaraj Murali, Bharath Sudheendra, Mark Buckler, Einar Veizaga, Shreesha Srinath, and Taylor Pritchard under the supervision of Chris Batten. Jiyue Yang also helped with the de-

sign and layout of ASP depth field pixels, and Gerd Kiene designed the amplifier presented in the Appendix B.

Special thanks go to Achuta Kadambi for many long discussions about computational imaging. I also enjoyed being a fly on the wall of the Cornell Graphics/Vision group especially chatting with Kevin Matzen, Scott Wehrwein, Paul Upchurch, Kyle Wilson, Sean Bell.

My heartfelt gratitude goes to my friends and family who supported me while I undertook this journey. I couldn't have done it without Buddy Rieger, Crystal Lee Morgan, Brandon Hang, Chris Hazel, Elly Engle, Steve Miller, Kevin Luke, Ved Gund, Alex Ruyack, Jayson Myers, Ravi Patel, and John McKay. Finally, I'd like to thank my brother Gihan and my parents for their love and support for all these years. This thesis is dedicated to them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Since the earliest recorded cave paintings such as those at Lascaux in modern-day France, human beings have tried to capture and reproduce their visual experiences. Such primitive capture methods were used practically for communication and scientific inquiry as well as aesthetic entertainment. The invention of the camera helped usher in the age of photography where images could be easily captured without specialized equipment or skills. Today, billions of photos are taken and shared online today, and is an integral part of our daily lives.

Using these photos, modern computer vision has emerged as a powerful tool to analyze information embedded in images. Researchers have focused on a variety of applications including 3D reconstruction from multiple 2D images of an object [53], image segmentation [51], tracking [213], and higher level semantic tasks such as object detection, recognition, and scene understanding [60, 156]. These algorithms have several use cases in robotics, industrial monitoring/detection, human-computer interaction, and entertainment which are enabled by digital photography.

Yet a photo is a low dimensional sampling of the plenoptic function of light, a function which outputs the radiance of a light ray traveling through a scene. The plenoptic function has multiple dimensions including space, angle, time, polarization, and wavelength. A 2D camera sensor captures a slice of this plenoptic function to create the photographs we see. A central theme in this thesis is that by sampling and processing additional plenoptic dimensions, we can extract even more information for visual computing algorithms than traditional digital photography.

To sample these additional plenoptic dimensions, we utilize computational cameras composed of Angle Sensitive Pixels (ASPs), photodiodes with integrated per-pixel diffraction gratings manufactured in a modern complementary metal-oxide semiconductor (CMOS) process. Prior work in ASPs have demonstrated the feasibility of designing and building these sensors, and deployed them for tasks such as incidence angle detection, edge filtering and image compression, and depth mapping.

In this thesis, we extend ASPs to perform plenoptic imaging by simultaenously capturing multiple plenoptic dimensions. We model the forward capture process for these dimensions, and develop new algorithms based on signal processing, machine learning, and computer vision from this raw data. For many applications, we weigh the relative advantages of using a single ASP camera against the associated disadvantages of reduced sampling resolution and lower signal-to-noise ratio (SNR) in each plenoptic dimension. Performing multimodal plenoptic imaging with our hardware platform, we showcase a variety of visual effects including looking past partial and scattering occluders, imaging stress in plastics, removing specular glints/highlights in a scene, improved depth mapping, and novel view synthesis. In addition, we show how ASP's plenoptic information can improve the energy efficiency of visual recognition from deep learning, a preliminary step towards true plenoptic computer vision. We hope that this thesis inspires more work in plenoptic imaging, and fundamentally changes how light is captured and processed in modern visual computing.

1.1 Dissertation Overview

The rest of this dissertation details how Angle Sensitive Pixels can capture and extract information from multiple dimensions of the plenoptic function of light as follows:

- Chapter 2 introduces relevant background on the plenoptic function of light and outlines existing ways to capture it. Angle Sensitive Pixels (ASPs) are introduced including previous research in design, fabrication, and signal processing for these sensors.
- Chapter 3 shows how ASPs can capture the angular dimensions of the plenoptic function, and presents a dictionary-based machine learning algorithm to recover high resolution 4D light fields. In addition, a new deep learning network is designed that achieves comparable visual quality to the dictionary-based method but improves reconstruction times by 5x or greater.
- Chapter 4 characterizes ASPs' polarization response and shows applications in imaging stress-induced birefringence and specular highlight removal from ASP depth mapping.
- Chapter 5 explores the feasibility of capturing time-of-flight information with ASPs. Depth fields are introduced as joint representations of depth and spatio-angular coordinates, and used to image past occlusion and resolve depth ambiguities. Preliminary designs for new CMOS depth field sensors are proposed as well.
- Chapter 6 leverages ASPs' edge filtering to perform optical computing for convolutional neural networks, saving energy for a modern computer vision pipeline.
- Chapter 7 summarizes conclusions from the previous chapters and points to future research directions for plenoptic imaging and ASPs.

In addition, we present two technical appendices on additional work in computational imaging. Appendix A presents a study on deep learning using binary gradient cameras, both analyzing the tradeoff between accuracy and power savings for different vision tasks as well as reconstructing gray level intensity images from binary

gradient images using autoencoders. Appendix B presents methodological work and tool development that makes the design and fabrication of computational image sensors easier and more robust to design errors. We present this as a tool for the research community to enable vertically-oriented research in the software/hardware stack for visual computing.

CHAPTER 2

BACKGROUND

This chapter covers the history of the plenoptic function of light, its modern formulation in computer vision and graphics, and surveys recent algorithms that exploit these different dimensions. We then survey the variety of computational cameras developed, and focus on Angle Sensitive Pixels in particular as the main hardware platform used in this thesis.

2.1 Plenoptic Function of Light

The first historically significant theory of light traveling as straight rays comes from Euclid in a treatise on geometric optics [22]. While a simplistic view of light transport, ray optics has enabled the invention of lenses, mirrors, telescopes, and cameras. In addition, much research in computer graphics and vision uses ray optics at the core of their algorithms. While the duality of the particle and wave nature of light can yield interesting visual phenomenon (e.g. interference, dispersion, etc), in this thesis, we mostly use ray optics (with the noted exception of polarization) to simplify our analysis.

The plenoptic function was first introduced by Adelson and Bergen [1] as a function which describes the radiance of a ray of light traveling in 3D space. Formally, we define the plenoptic function as follows:

$$L(x, y, z, \theta, \phi, \lambda, t, \chi), \tag{2.1}$$

where output is radiance measured in units of Watts per steradian (solid angle) per meters squared (area). Note that radiance differs from irradiance which is measured in power per area, and both are sometimes colloquially referred to as "intensity" even though intensity

can be formally defined as power per steradian. We refer the reader to the work by Nicodemus [144] for precise definitions of these terms with respect to both received and emitted light.

The formulation above parameterizes a ray by the following variables: (x, y, z) for the 3D position of a ray, (θ, ϕ) for the direction, λ for the wavelength or color, t for time, and χ for the polarization state. This formulization is not canonical as many researchers limit the plenoptic function to omit λ, t, do not include z when a ray is traveling in unoccluded space, or ignore polarization (technically a wave effect of light). In addition, our formulation does not consider further effects such as coherency of light or diffraction. However, this thesis shows that this parameterization of the plenoptic function can yield interesting, novel visual computing applications.

Changes in any of the above variables can yield changes in radiance for a ray, and this plenoptic function is dependent on both scene geometry and light transport in that scene. A goal of plenoptic imaging is to infer information about the scene geometry and lighting from sampling this plenoptic function. This is a natural extension of image-based rendering [134] in computer graphics to incorporate more plenoptic dimensions. In the next section, we discuss what previous research has accomplished for this task.

2.2 Applications of Plenoptic Imaging

Plenoptic imaging necessitates two steps: capturing the plenoptic function and then inverting the representation to infer scene/lighting properties. We discuss the latter step in this section. Readers interested in imaging devices to capture the plenoptic function should start at Section 2.3. Note that we skip discussing the dimension of space (x, y, z) since this

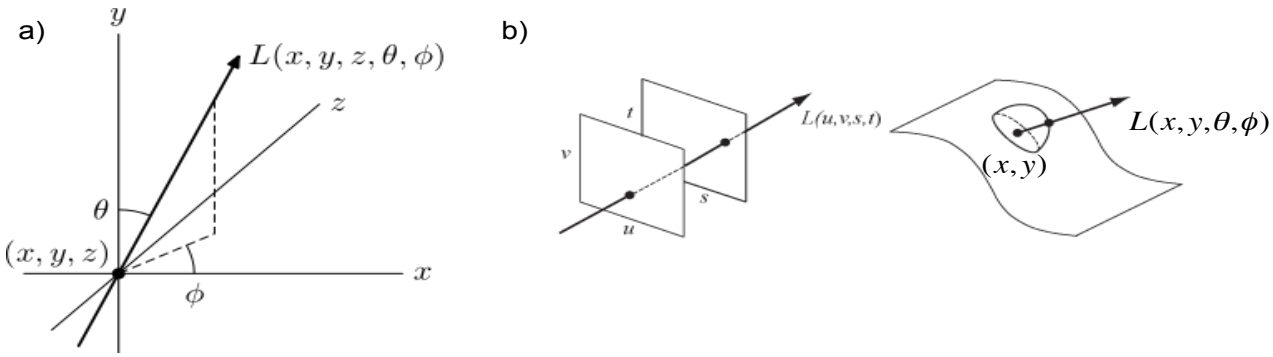


Figure 2.1: a) The radiance of a ray in 3D space as function of spatio-angular dimensions. This is commonly known as the 5D light field. b) Common parameterizations for 4D Light Fields. Source: https://en.wikipedia.org/wiki/Light_field

is the domain of general photography and computer vision, and has been heavily surveyed by others (See Computer Vision: A Modern Approach [52] and Multiple View Geometry in Computer Vision [74] for a general overview).

Angle: If we only consider (x, y, z, θ, ϕ) for the plenoptic function, we can describe the set of rays in 3D space collectively known as a 5D light field. Further assuming that the rays are traveling in unoccluded space, we can simplify to (x, y, θ, ϕ) as a 4D light field. 4D light fields were introduced independently by Levoy and Hanrahan [121] and Gortler et al. (called the lumigraph) [65]. There are multiple ways to parameterize this 4D light field, but the two most common are the two-plane parametrization (x, y, u, v) and the angular (x, y, θ, ϕ). See Figure 2.1 for a visual depiction of common light field parameterizations.

Since their introduction, 4D light fields have seen numerous uses in computer vision and graphics. Light fields have been used for synthesizing images from novel viewpoints, matting and compositing, image relighting, and computational refocusing. Several algorithms for recovering depth from light fields have been proposed, using defocus cues as well as epipolar geometry [179]. This depth has been useful for segmentation and recovering 3D geometry [217].

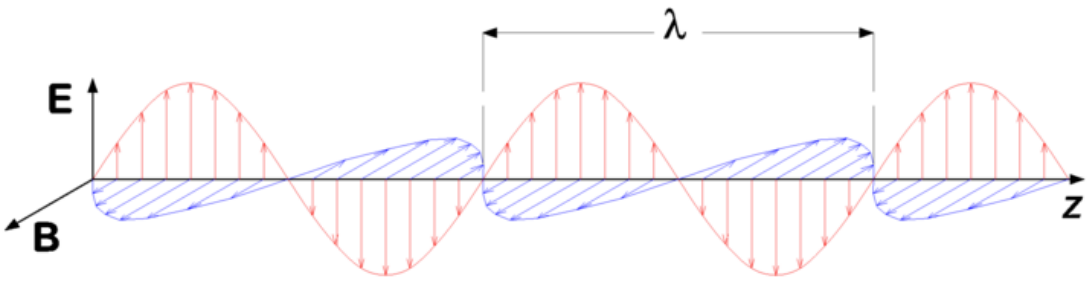


Figure 2.2: A visualization of linearly polarized light. Source: https://en.wikipedia.org/wiki/Polarization_(waves)

Polarization: Polarization is the orientation of the electromagnetic wave of light (see Fig 2.2), and changes upon light interactions (such as reflection or transmission) with matter. This is useful for inferring physical properties of materials in a scene. A formal description of polarization involves Stokes' vectors and Mueller matrices to describe polarized light transport [78]. However, for most practical applications in computer vision and graphics, researchers have dealt with simplified physical models for specular and diffuse reflection [183] and scattering [160].

Main applications of polarization imaging include material indentification [205], imaging through haze, fog, and underwater [159, 158], and even biomedical endoscopy of cancerous tissue [215]. Since polarized reflection also gives you information about surface normals, shape from polarization has been a popular computer vision algorithm to recover shape of 3D objects [94, 7]. However, it should be noted that polarization is still underutilized in modern vision and graphics due to the inaccessiblity of data and light transport simulators.

Time-of-Flight: While not strictly a plenoptic dimension, time-of-flight (TOF) uses temporal changes in radiance to calculate the depth of an object. This does require actively

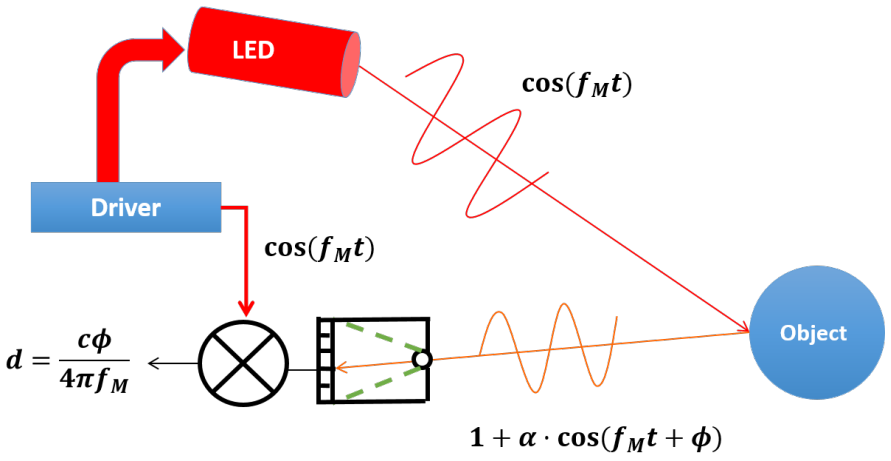


Figure 2.3: Continuous-wave time-of-flight imaging measures the phase shift due to optical path length of time-varying illumination. This phase shift is proportional to the distance traveled by the light.

illuminating the scene to change the reflected radiance. There are two main types of time-of-flight illumination, pulsed or continuous wave. A short pulse of light requires fast detectors that can resolve time differences on the order of picoseconds for reasonable depth accuracy. In contrast, continuous wave TOF sinusoidally modulates a light source at a given frequency. This light aquires a phase shift corresponding to the optical path length traveled, which is then decoded at the sensor by using a correlation sensor [109] to measure the phase shift. Depth can calculated from this phase shift (see Fig 2.3 and Ch. 5 for the derivation of this formula).

TOF technology has achieved widespread commercial success, appearing as the 3D sensor in systems including the Microsoft Kinect, Hololens, and Google's Project Tango. Researchers have improved TOF to perform transient imaging of light-in-flight [187], handle multipath interference [95], and even material recognition [171]. As solid-state LIDARs/LEDs and the associated detectors scale with technology to be cheaper and smaller, its a safe bet to imagine RGB-D cameras ubiquitously deployed in the future.

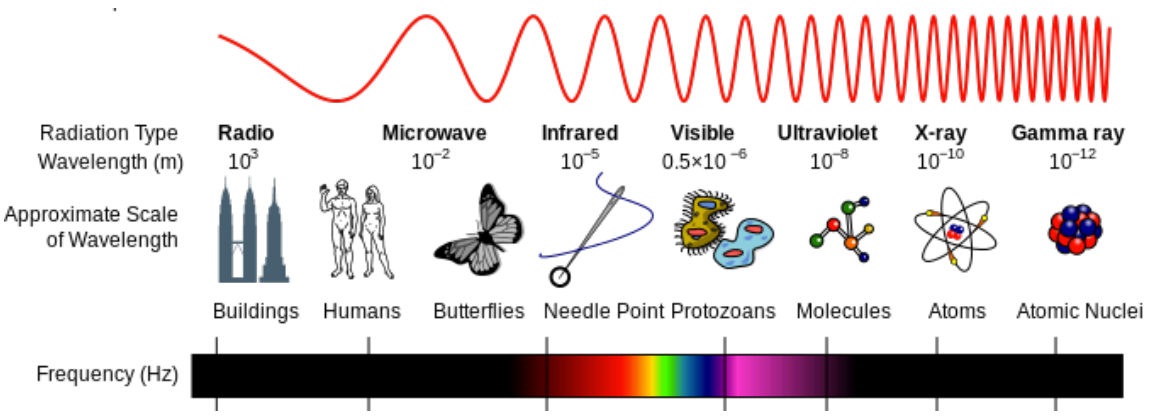


Figure 2.4: Wavelengths of the electromagnetic spectrum. CMOS image sensors are only sensitive from visible light to near infrared (1100nm). Source: https://en.wikipedia.org/wiki/Electromagnetic_spectrum

Wavelength: Hyperspectral imaging is another area of imaging that uses information about wavelength to infer properties of a scene and materials. The most common type of hyperspectral imaging is the Bayer pattern of RGB bandpass filters used on top of CMOS image sensors for color [12]. Capturing finer wavelengths within the detection range of silicon (300nm to about 1100nm) requires optical filters involving dielectric coatings or optical elements such as diffraction grating. In addition, new detectors are being developed to extend the available spectrum that can be captured such as infrared and x-ray.

There has been much research in using color cues for computer vision [139], color science [see [208] for a general overview], and even capturing hypercubes of spectral data [57]. However, in this thesis we do not address wavelength primarily due to the fact that we use monochrome CMOS image sensors with no embedded Bayer filters. This does remain an interesting avenue for future research.

2.3 Computational Cameras

In many senses, every generation of camera has incorporated more computation and processing to produce the final photograph. Switching from films to CCD and CMOS image sensors [47] and the development of the image sensor processor (ISP) [154] with algorithms such as white balancing, color mapping, gamma correction, and demosiacking, computation has played a central role in making photographs more visually appealing. We survey the subset of computational cameras designed to capture more dimensions of the plenoptic function of light.

4D light fields have been captured by camera arrays [204, 188], spherical gantries, and robotic arms. Early prototypes of single-shot light field cameras either used a microlens array [129] or a light-blocking mask [87] to multiplex the rays of a 4D light field onto a 2D sensor. In the last decades, significant improvements have been made to these basic designs, i.e. microlens-based systems have become digital [2, 141] and mask patterns more light efficient [186, 111]. Recently, light fields have been captured through even more exotic optical elements including diffusers [5] and even water droplets [201].

Capturing polarization has primarily involved the use of external polarizing filters for cameras. This include manually rotating a polarizing filter as well as mechanical rotors or LCDs [206]. To detect polarization without these external optics, researchers have used fabricated nanowires [70], aligned structures [69], or integrated metal interconnects [157] to create polarizing filters that tile the image sensor known as division of focal plane polarizers.

For time-of-flight, researchers have developed pulse based detectors from streak cameras [176] and single photon avalanche diodes [37]. For continuous wave, time-of-

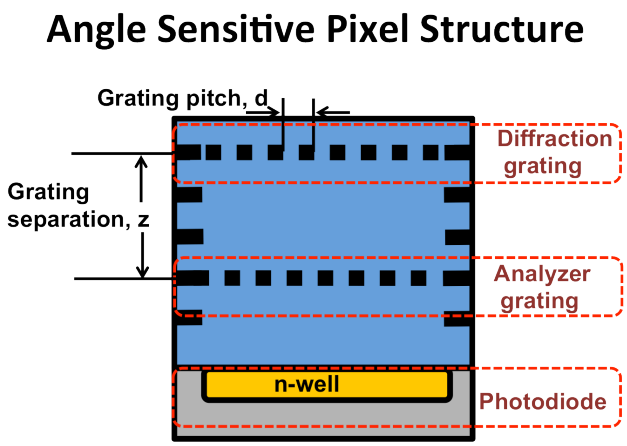


Figure 2.5: The design of an Angle Sensitive Pixel (ASP) [196]

flight pixels include photogates, photonic mixer devices, and lateral electric field modulators [9, 62, 110, 162, 98]. Currently, scanning LIDAR systems are being deployed in autonomous vehicles including Google and Uber.

2.4 Angle Sensitive Pixels

As stated before, our main hardware platform to perform plenoptic imaging are image sensors composed of Angle Sensitive Pixels (ASPs). In this section, we survey previous research on ASPs including both advances in design/fabrication as well as applications. At the end, we outline how this thesis establishes ASPs as multimodal plenoptic image sensors and the resulting applications.

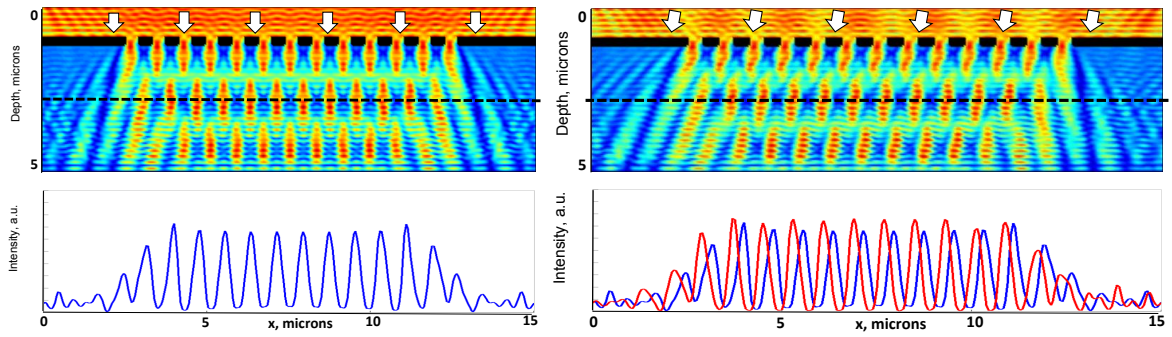


Figure 2.6: Plane waves of light impinging on a diffraction grating yield a sinusoidal interference effect which shifts laterally as a function of incidence angle (k vector) [192].

2.4.1 Background

ASPs are photodiodes, typically implemented in a CMOS fabrication process, with integrated diffraction gratings on the order of the wavelength of visible light (500nm - 1μm) [192], see Figure 2.5 for the pixel cross-section. When a plane wave of light hits the first diffraction grating, it exhibits an interference pattern known as the Talbot effect, a periodic self-image of the grating at characteristic depths known as Talbot depths (See Figure 2.6). This pattern shifts laterally as a function of the incoming incidence angle of the wave, and thus imaging how the pattern shifts would recover incidence angle (up to 2π). However, these shifts are on the order of nanometers, which is much smaller than the photodiode size of current technology (1.1μm).

To overcome this issue, ASPs use a second set of diffraction gratings (called the analyzer grating) to filter the interference pattern selectively with respect to incidence angle (See Figure 2.7. This gives a characteristic sinusoidal response to incidence angle of light, but requires multiple pixels to resolve ambiguities in the measurement [192]. ASPs extend this response to 2D incidence angle by using grating orientation, pitch, and relative phase

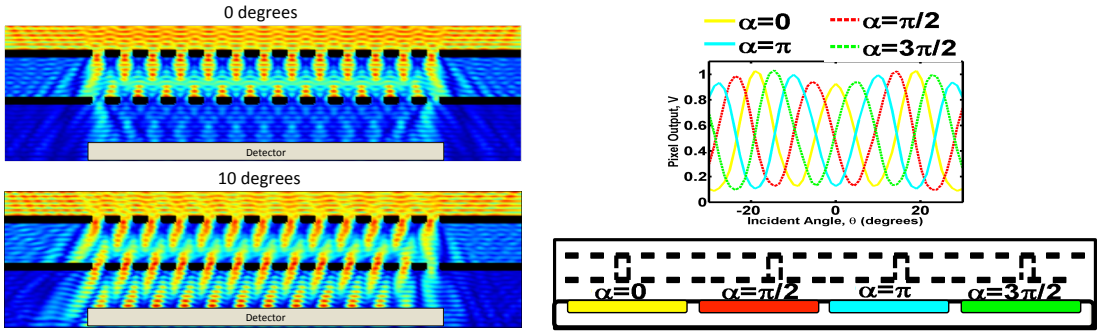


Figure 2.7: The second grating, known as the analyzer grating, helps selectively filter certain incidence angles of light from arriving at the photodetector underneath. Four different phases of analyzer gratings relative to the diffraction gratings give a quadrature response to incidence angle [196]

of the diffraction to analyzer grating. This response is given by the following equation [84]:

$$i(x,y) = 1 + m\cos\left(\beta\left(\cos\left(\gamma\right)\theta_x + \sin\left(\gamma\right)\theta_y\right) + \alpha\right), \tag{2.2}$$

where θ_x, θ_y are 2D incidence angles, α, β, γ are parameters of the ASP pixel corresponding to phase, angular frequency, and grating orientation, and m is the amplitude of the response. See [168] for further details of how to design gratings in a CMOS process and optimize parameter selection.

A tile of ASPs contains a diversity of 2D angle responses, and is repeated periodically over the entire image sensor to obtain multiple measurements of the local light field. Several chip variations of ASPs have been presented in [192, 196, 190], please refer to these papers for information on designing ASP arrays, readout circuitry, and digitization. An advantage of these sensors is that they are fully CMOS-compatible and thus can be manufactured in a low-cost industry fabrication process.

Improvements in Pixel Technology: ASPs have been known to suffer from limitations including loss of light due to two diffraction gratings and the need for multiple pixels to resolve angle ambiguity. These limitations do show up in subsequent chapters as practical

difficulties for certain applications. However, recent work in ASP pixel fabrication has introduced phase gratings to increase relative quantum efficiency up to 50% , and introduced interleaved photodiode designs to increase spatial density by 2× [169, 168]. Future sensors with these pixels will hopefully alleviate some of the limitations for ASP imaging.

2.4.2 Applications

Since their introduction, ASPs have been used for several imaging applications. Incident angle sensitivity has been used for localizing fluorescent sources without a lens [191] and measuring changes in optical flow [195]. Characterizing ASP optical impulse responses as Gabor filters (or oriented spatial frequency filters) [194, 196], researchers have used these filters to show computational refocusing and depth mapping without using an explicit light field formulation. In addition, ASPs have been designed to compress images using edge filtering [190], and arrays of ASPs can act as lensless imagers [59].

2.4.3 Our Contribution

While many of the previous applications of ASPs are useful for specific tasks, it is the aim of this thesis to present ASP imaging under a unified plenoptic imaging framwork. This approach presents a common, extensible forward model for light capture for multiple plenoptic dimensions and allows us to compare ASPs with other plenoptic cameras. In Ch. 3-5 of the thesis, we present a forward model for capturing a new plenoptic dimension and show applications when we invert this model for scene inference. Ch. 6 extends a property of ASP optical edge filtering to perform optical convolution of a convolutional neural network for energy-efficient deep learning. All of these advances showcase ASPs

and the the promise of multimodal plenoptic imaging in general.

CHAPTER 3

ANGLE

Angle Sensitive Pixels are limited by a spatio-angular tradeoff common to all single-shot light field cameras when capturing 4D light fields. To overcome this tradeoff, we utilize recent techniques from machine learning and sparsity-based optimization to recover the missing spatial resolution for the light fields. As a result of our algorithms[1], we obtain high resolution 4D light fields captured from a prototype ASP camera, and showcase common light field applications including view synthesis and computational refocusing.

However this dictionary-based algorithm can take up to several hours to achieve good visual quality when processing these light fields. We introduce a new neural network architecture[2] that reconstructs 4D light fields at higher visual quality, and improves reconstruction times to a few minutes, pointing to the potential of achieving real-time light field video with ASPs.

3.1 Introduction

Over the last few years, light field acquisition has become one of the most widespread computational imaging techniques. By capturing the 4D spatio-angular radiance distribution incident on a sensor, light field cameras offer an unprecedented amount of flexibility for data processing. However, conventional light field cameras are subject to the spatio-angular resolution tradeoff. Whereas angular light information is captured to enable a variety of new modalities, this usually comes at the cost of severely reduced image resolu-

[1]This work was originally presented in M. Hirsch et al., "A switchable light field camera architecture using Angle Sensitive Pixels and dictionary-based sparse coding", ICCP 2014 [84].

[2]This work was originally presented in M. Gupta et al., "Compressive light field reconstructions using deep learning" (submitted to CVPR 2017)

tion. Recent efforts have paved the way for overcoming the resolution tradeoff using sparse coding [135] or super-resolution techniques [188, 20]. Although these methods slightly improve the resolution of 4D light fields, it is still significantly lower than that offered by a regular camera sensor with the same pixel count. One may argue that light field cameras would be most successful if they could seamlessly switch between high-resolution 2D image acquisition and 4D light field capture modes.

In this chapter, we explore such a switchable light field camera architecture. We combine ASP hardware with modern techniques for compressive light field reconstruction and other processing modes into what we believe to be the most flexible light field camera architecture to date.

In particular, we make the following contributions:

- We present a switchable camera allowing for high-resolution 2D image and 4D light field capture. These capabilities are facilitated by combining ASP sensors with modern signal processing techniques.
- We analyze the imaging modes of this architecture and demonstrate that a single image captured by the proposed camera provides either a high-resolution 2D image using little computation, a medium-resolution 4D light field using a moderate amount of computation, or a high-resolution 4D light field using more compute-intense compressive reconstructions.
- We evaluate system parameters and compare the proposed camera to existing light field camera designs. We also show results from a prototype camera system.

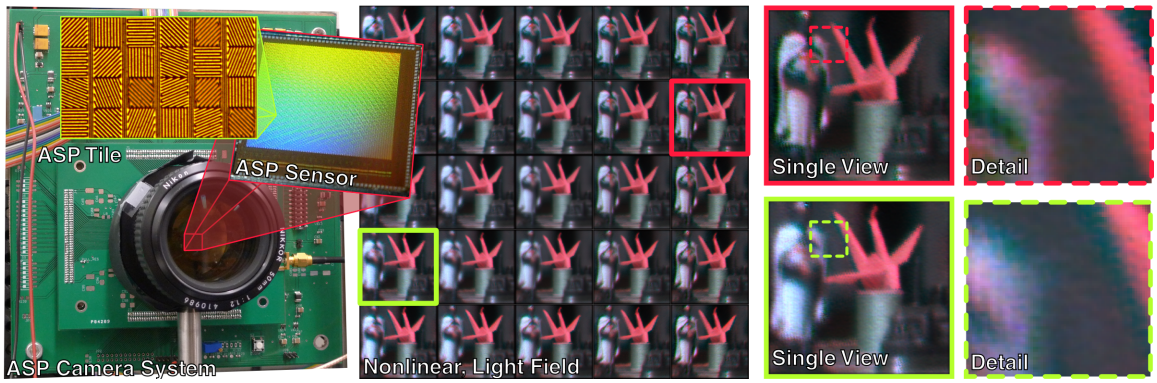


Figure 3.1: Prototype angle sensitive pixel camera (left). The data recorded by the camera prototype can be processed to recover a high-resolution 4D light field (center). As seen in the close-ups on the right, parallax is recovered from a single camera image.

3.2 Related Work

It is well-understood that light fields of natural scenes contain a significant amount of redundancy. Most objects are diffuse; a textured plane at some depth, for instance, will appear in all views of a captured light field, albeit at slightly different positions. This information can be fused using super-resolution techniques, which compute a high-resolution image from multiple subpixel-shifted, low-resolution images [177, 164, 17, 132, 150, 188, 20, 198].

With the discovery of compressed sensing [23, 41], a new generation of compressive light field camera architectures is emerging that goes far beyond the improvements offered by super-resolution. For example, the spatio-angular resolution tradeoff in single-device light field cameras [6, 8, 210, 135] can be overcome or the number of required camera in arrays reduced [97]. Compressive approaches rely on increased computational processing with sparsity priors to provide higher image resolutions than otherwise possible.

The camera architecture proposed in this chapter is well-suited for compressive re-

Technique	Light Transmision	Image Resolution	Single Shot	Single Device	Comp. Complexity	High-Res 2D Image
Microlenses	high	low	yes	yes	low	no
Pinhole Masks	low	low	yes	yes	low	no
Coded Masks (SoS, MURA)	medium	low	yes	yes	medium	no
Scanning Pinhole	low	high	no	yes	low	yes
Camera Array	high	high	yes	no	medium	yes
Compressive LF	medium	high	yes	yes	high	yes*
Proposed Method	high	low / high	yes	yes	medium / high	yes

*With extra computation

Table 3.1: Overview of benefits and limitations of light field photography techniques. As opposed to existing approaches, the proposed computational camera system provides high light field resolution from a single recorded image. In addition, our switchable camera is flexible enough to provide additional imaging modes that include conventional, high-resolution 2D photography.

constructions, for instance with dictionaries of light field atoms [135]. In addition, our approach is flexible enough to allow for high-quality 2D image and lower-resolution light field reconstruction from the *same measured data* without numerical optimization.

3.3 Method

This section introduces the image formation model for ASP devices. In developing the mathematical foundation for these camera systems, we entertain two goals: to place the camera in a framework that facilitates comparison to existing light field cameras, and to understand the plenoptic sampling mechanism of the proposed camera.

3.3.1 Light Field Acquisition with ASPs

The Talbot effect created by periodic grating induces a sinusoidal angular response for angle sensitive pixels (ASPs). For a one-dimensional ASP, this can be described as

$$\rho^{(\alpha,\beta)}(\theta) = (1 + m\cos(\beta\theta + \alpha)) \,. \tag{3.1}$$

Here, α and β are phase and frequency, respectively, m is the modulation efficiency, and θ is the angle of incident light. Both α and β can be tuned in the sensor fabrication process [193]. Common implementations choose ASP types with $\alpha \in 0, \pi/2, \pi, 3\pi/4$.

Similarly, 2D ASP implementations exhibit the resulting angular responses for incident angles θ_x and θ_y:

$$\rho^{(\alpha,\beta,\gamma)}\left(\boldsymbol{\theta}\right) = 1 + m\cos\left(\beta\left(\cos\left(\gamma\right)\theta_x + \sin\left(\gamma\right)\theta_y\right) + \alpha\right), \tag{3.2}$$

where α is phase, β is frequency, and γ is grating orientation.

The captured sensor image i is then a projection of the incident light field l weighted by the angular responses of a mosaic of ASPs:

$$i\left(\mathbf{x}\right) = \int_{\mathcal{V}} l(\mathbf{x}, \boldsymbol{\nu})\ \rho\left(\mathbf{x}, \tan^{-1}(\boldsymbol{\nu})\right)\ \omega\left(\boldsymbol{\nu}\right)\ d\boldsymbol{\nu} \,. \tag{3.3}$$

In this formulation, $l(\mathbf{x}, \boldsymbol{\nu})$ is the light field inside the camera behind the main lens. We describe the light field using a relative two-plane parameterization [46], where $\boldsymbol{\nu} = \tan(\boldsymbol{\theta})$. The integral in Equation 3.3 contains angle-dependent vignetting factors $\omega\left(\boldsymbol{\nu}\right)$ and the aperture area $\mathcal{V}$ restricts the integration domain. Finally, the spatial coordinates $\mathbf{x} = \{x, y\}$ are defined on the sensor pixel-level; the geometrical microstructure of ASP gratings and photodiodes is not observable at that scale. In practice, the spatially-varying pixel response function $\rho\left(\mathbf{x}, \boldsymbol{\theta}\right)$ is a periodic mosaic of a few different ASP types. A common example

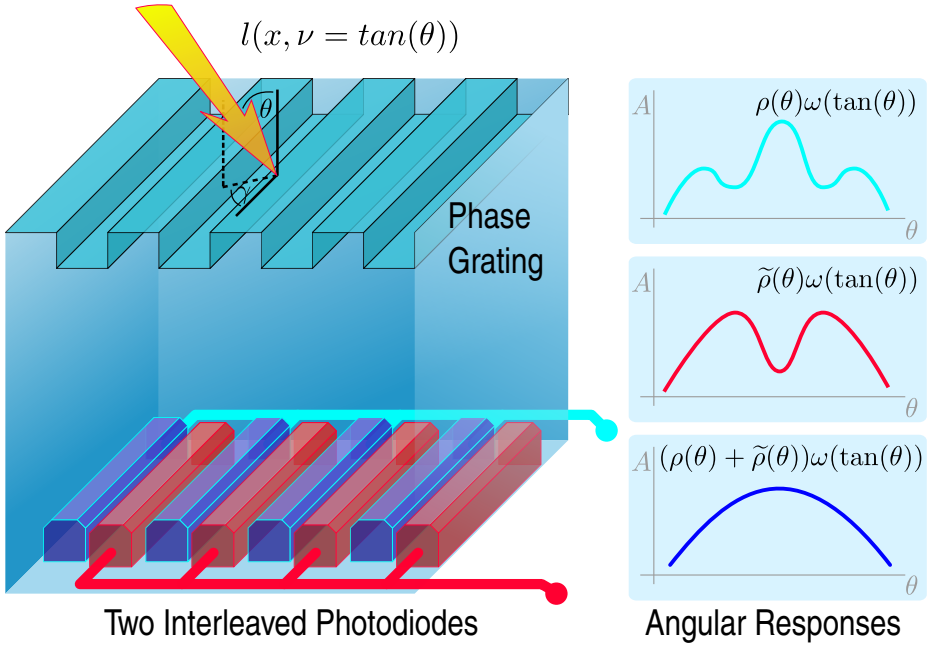


Figure 3.2: Schematic of a single angle sensitive pixel. Two interleaved photodiodes capture a projection of the light field incident on the sensor (left). The angular responses of these diodes are complementary: a conventional 2D image can be synthesized by summing their measurements digitally (right).

of such a layout for color imaging is the Bayer filter array that interleaves red, green, and blue subpixels. ASPs with different parameters (α, β, γ) can be fabricated following this scheme. Mathematically, this type of spatial multiplexing is formulated as

$$\rho\left(\mathbf{x}, \boldsymbol{\theta}\right) = \sum_{k=1}^{N} \left(\text{Ш}^{(k)}(\mathbf{x}) * \rho^{(\zeta(k))}\left(\boldsymbol{\theta}\right)\right), \tag{3.4}$$

where $*$ is the convolution operator and $\text{Ш}^{(k)}(\mathbf{x})$ is a sampling operator consisting of a set of Dirac impulses describing the spatial layout of one type of ASP. A total set of N types is distributed in a regular grid over the sensor. The parameters of each are given by the mapping function $\zeta(k) : \mathbb{N} \rightarrow \mathbb{R}^3$ that assigns a set of ASP parameters (α, β, γ) to each index k.

Whereas initial ASP sensor designs use two layered, attenuating diffraction gratings and conventional photodiodes underneath [192, 193, 59], more recent versions enhance the quantum efficiency of the design by using a single phase grating and an interleaved pair of photodiodes [169]. For the proposed switchable light field camera, we illustrate the latter design with the layout of a single pixel in Figure 3.2.

In this sensor design, each pixel generates two measurements: one that has an angular response described by Equation 6.1 and another one that has a complementary angular response $\widetilde{\rho} = \rho^{(\alpha+\pi,\beta,\gamma)}$ whose phase is shifted by π. The discretized version of the two captured images can be written as a simple matrix-vector product:

$$\mathbf{i} = \mathbf{\Phi l}, \tag{3.5}$$

where $\mathbf{i} \in \mathbb{R}^{2m}$ is a vector containing both images $i\left(\mathbf{x}\right)$ and $\widetilde{i}\left(\mathbf{x}\right)$, each with a resolution of m pixels, and $\mathbf{\Phi} \in \mathbb{R}^{2m} \times \mathbb{R}^{n}$ is the projection matrix that describes how the discrete, vectorized light field $\mathbf{l} \in \mathbb{R}^{n}$ is sensed by the individual photodiodes.

3.3.2 Image and Light Field Synthesis

In this section, we propose a number of alternative ways to process the data recorded with an ASP sensor.

Direct 2D Image Synthesis: As illustrated in Figure 3.2, the angular responses of the complementary diodes in each pixel can simply be summed to generate a conventional 2D image, i.e. $\rho^{(\alpha,\beta,\gamma)} + \widetilde{\rho}^{(\alpha,\beta,\gamma)}$ is a constant. Hence, Equation 3.3 reduces to the conventional photography equation:

$$i\left(\mathbf{x}\right) + \widetilde{i}\left(\mathbf{x}\right) = \int_{\mathcal{V}} l(\mathbf{x}, \boldsymbol{\nu})\, \omega\left(\boldsymbol{\nu}\right)\, d\boldsymbol{\nu}, \tag{3.6}$$

which can be implemented in the camera electronics. Equation 3.6 shows that a conventional 2D image can easily be generated from an ASP sensor. While this may seem trivial, existing light field camera architectures using microlenses or coded masks cannot easily synthesize a conventional 2D image for in-focus and out-of-focus objects.

Linear Reconstruction for Low Resolution 4D Light Fields: Using a linear reconstruction framework, the same data can alternatively be used to recover a low-resolution 4D light field. We model light field capture by an ASP sensor as Equation 3.5 where the rows of $\mathbf{\Phi}$ correspond to vectorized 2D angular responses of different ASPs. These angular responses are either sampled uniformly from Equation 6.1 or they can be fitted empirically from measured impulses responses. The approximate orthonormality of the angular wavelets (see Sec. 3.5) implies $\mathbf{\Phi}^T\mathbf{\Phi} \approx \boldsymbol{I}$, which we consequently use $\mathbf{\Sigma} = \text{diag}(\mathbf{\Phi}^T\mathbf{\Phi})$ as a preconditioner for inverting the capture equation: $\mathbf{l} = \mathbf{\Sigma}^{-1}\mathbf{\Phi}^T\mathbf{i}$.

The main benefit of a linear reconstruction is its computational performance. However, the spatial resolution of the resulting light field will be approximately k-times lower than that of the sensor ($k = n/m$) since the different ASPs are grouped into tiles on the sensor.

Similar to demosaicing for color filter arrays, demosaicing different angular measurements for the ASP sensor design can be done using interpolation and demultiplexing as described in [202] to yield better visual resolution. In addition, recent work on light field super-resolution has demonstrated that this resolution loss can also be slightly mitigated for the particular applications of image refocus [188] and volume reconstruction [20].

Sparse Coding for High-Resolution Light Fields: Finally, we can choose to follow Marwah et al. [135] and apply nonlinear sparse coding techniques to recover a high-resolution 4D light field from the same measurements. This is done by representing the light field using an overcomplete dictionary as $\mathrm{l} = \boldsymbol{\mathcal{D}}\boldsymbol{\chi}$, where $\boldsymbol{\mathcal{D}} \in \mathbb{R}^{n \times d}$ is a dictionary of light field atoms and $\boldsymbol{\chi} \in \mathbb{R}^{d}$ are the corresponding coefficients. Natural light fields have been shown to be sparse in such dictionaries [135], i.e. the light field can be represented as a weighted sum of a few light field atoms (columns of the dictionary). For robust reconstruction, a basis pursuit denoise problem (BPDN) is solved

$$\begin{aligned} \underset{\{\boldsymbol{\chi}\}}{\text{minimize}} \quad & \|\boldsymbol{\chi}\|_1 \\ \text{subject to} \quad & \|\mathbf{i} - \boldsymbol{\Phi}\boldsymbol{\mathcal{D}}\boldsymbol{\chi}\|_2 \leq \epsilon, \end{aligned} \tag{3.7}$$

where ϵ is the sensor noise level. Whereas this approach offers significantly increased light field resolution, it comes at an increased computational cost. Note that Equation 3.7 is applied to a small, sliding window of the recorded data, each time recovering a small 4D light field patch rather than the entire 4D light field at once. In particular, window blocks with typical sizes of 8×8 pixels are processed in parallel to yield light field patches with $8 \times 8 \times 5 \times 5$ rays each. Please see Section 3.5.2 for implementation details.

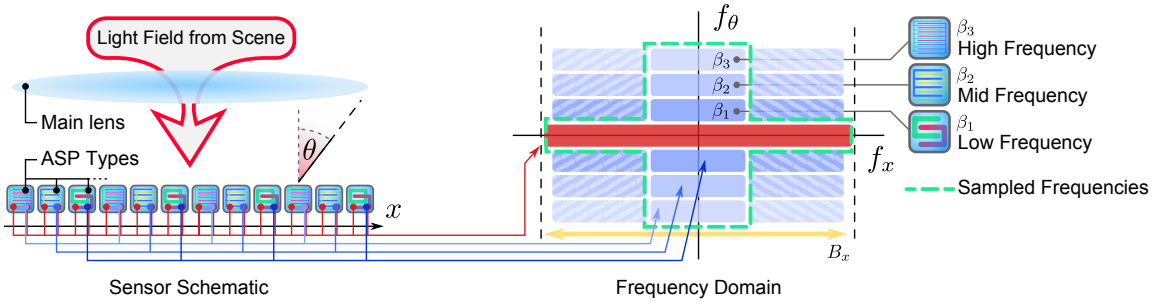


Figure 3.3: Illustration of ASP sensor layout (left) and sampled spatio-angular frequencies (right). This sensor interleaves three different types of ASPs. Together, they sample all frequencies contained in the dashed green box (right). A variety of light field reconstruction algorithms can be applied to these measurements, as described in the text.

3.4 Analysis

In this section, we analyze the proposed methods and compare them to alternative light field sensing approaches.

3.4.1 Frequency Analysis

As discussed in the previous section, Angle Sensitive Pixels sample a light field such that a variety of different reconstruction algorithms can be applied to the same measurements. To understand the information contained in the measurements, we can turn to a frequency analysis. Figure 3.3 (left) illustrates a one-dimensional ASP sensor with three interleaved types of ASPs sampling low, mid, and high angular frequencies, respectively. As discussed in Section 3.3, the combined measurements of the two interdigitated diodes in each pixel can be combined to synthesize a conventional 2D image. This image has no angular information but samples the entire spatial bandwidth B_x of the sensor (Fig. 3.3 right, red box).

The measurements of the individual photodiodes contain different angular frequency bands, but only for lower spatial frequencies due to the interleaved sampling pattern (Fig. 3.3 right, solid blue boxes). A linear reconstruction would require an optical anti-aliasing filter to be mounted on top of the sensor. These types of optical filters are part of most commercial sensors. In the absence of an optical anti-aliasing filter, aliasing is observed. For the proposed application, aliasing results in high spatio-angular frequencies (Fig. 3.3 right, dashed blue boxes) to be optically mixed into lower frequencies. The entire sampled spatio-angular frequencies are highlighted by the dashed green box (Fig. 3.3 right). Although aliasing makes it difficult to achieve high-quality reconstructions with simple linear demosaicing, it is crucial for nonlinear, high-resolution reconstructions based on sparsity-constained optimization.

3.4.2 Depth of Field

To evaluate the depth of field that can be achieved with the proposed sparsity-constrained reconstruction methods, we simulate a two-dimensional resolution chart at multiple different distances to the camera's focal plane. The results of our simulations are documented in Figure 3.4. The camera is focused at 50 cm, where no parallax is observed in the light field. At distances closer to the camera or farther away the parallax increases—we expect the reconstruction algorithms to achieve a lower peak signal-to-noise ratio (PSNR). The PSNR is measured between the depth-varying target 4D light field and the reconstructed light field.

Figure 3.4 (top) compares sparsity-constrained reconstructions using different measurement matrices and also a direct sampling of the low-resolution light field using microlenses (red plot). Slight PSNR variations in the latter are due to the varying size of the resolution

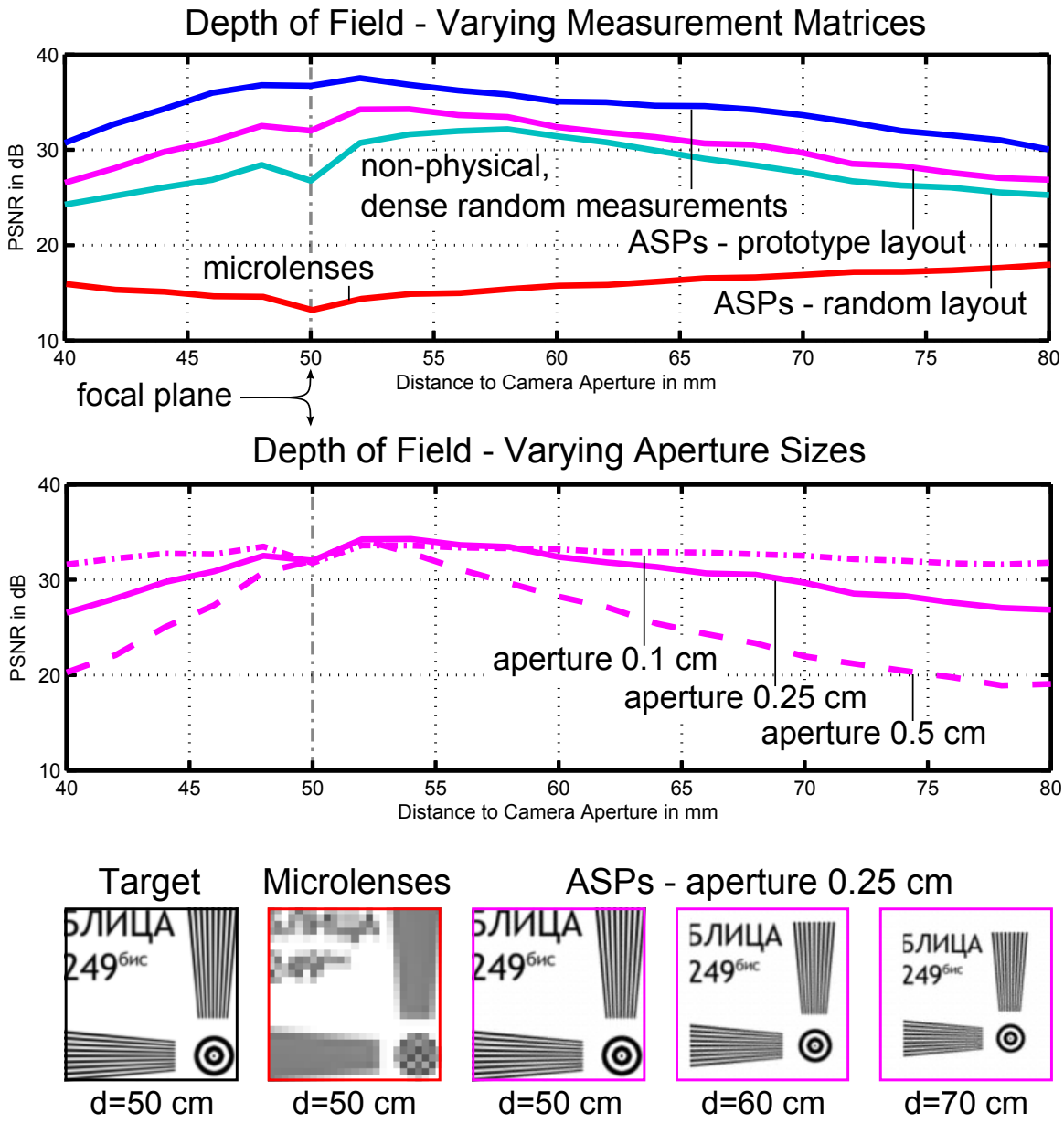


Figure 3.4: Evaluating depth of field. Comparing the reconstruction quality of several different optical setups shows that the ASP layout in the prototype camera is well-suited for sparsity-constrained reconstructions using overcomplete dictionaries (top). The dictionaries perform best when the parallax in the photographed scene is smaller or equal to that of the training light fields (center). Central views of reconstructed light fields are shown in the bottom.

chart in the depth-dependent light fields, which is due to the perspective of the camera (cf. bottom images). Within the considered depth range, microlenses always perform poorly. The different optical setups tested for the sparsity-constrained reconstructions include the ASP layout of our prototype (magenta plot, described in Sec. 3.5.1), ASPs with completely random angular responses that are also randomized over the sensor (green plot), and also a dense random mixing of all light rays in each of the light field patches (blue plot). Whereas the latter setup is physically not realizable, it gives us an intuition of the approximate upper performance bounds that could be achieved. Unsurprisingly, such a dense, random measurement matrix Φ performs best. What is surprising, however, is that random ASPs are worse than the choice of regularly-sampled angular wavelet coefficients in our prototype (see Sec. 3.5.1). For compressive sensing applications, the rows of the measurement matrix Φ should be as incoherent (or orthogonal) as possible to the columns of the dictionary $\mathcal{D}$. For the particular dictionary used in these experiments, random ASPs seem to be more coherent with the dictionary. These findings are supported by Figure 3.5. We note that the PSNR plots are content-dependent and also dependent on the employed dictionary.

The choice of dictionary is critical. The one used in Figure 3.4 is learned from 4D light fields showing 2D planes with random text within the same depth range as the resolution chart. If the aperture size of the simulated camera matches that used in the training set (0.25 cm), we observe high reconstruction quality (solid line, center plots). Smaller aperture sizes will result in less parallax and can easily be recovered as well, but resolution charts rendered at larger aperture sizes also contain a larger amount of parallax than any of the training data. The reconstruction quality in this case drops rapidly with increasing distance to the focal plane (Fig. 3.4, center plots).

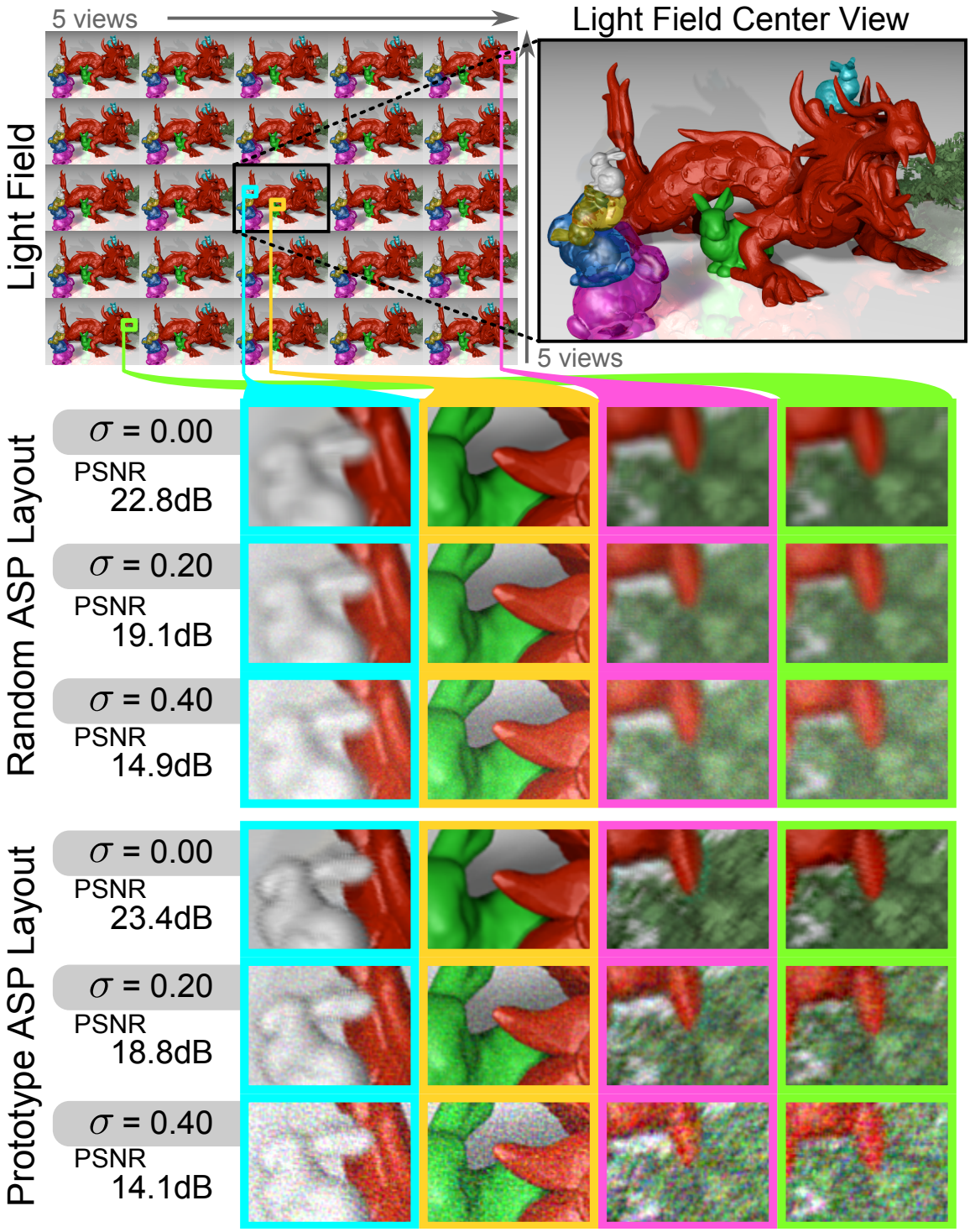


Figure 3.5: Simulated light field reconstructions from a single coded sensor image for different levels of noise and two different optical sampling schemes. For the ASP layout in the prototype camera (bottom), high levels of noise result in noisy reconstructions—parallax is faithfully recovered (dragon's teeth, lower right). A physically-realizable random ASP layout (center) does not measure adequate samples for a sparse reconstruction to recover a high-quality light field from a single sensor image. In this case, the reconstructions look blurry and parallax between the views is not recovered (center, right).

3.4.3 Resilience to Noise

Finally, we evaluate the sparse reconstruction algorithm w.r.t. noise and compare two different optical sampling schemes. Figure 3.5 shows a synthetic light field with 5×5 different views. We simulate sensor images with zero-mean i.i.d. Gaussian noise and three different standard deviations $\sigma = \{0.0, 0.2, 0.4\}$. In addition, we compare the ASP layout of the prototype (see Sec. 3.5.1) with a random layout of ASPs that each also have a completely random angular response. Confirming the depth of field plots in Figure 3.4, a random ASP layout achieves a lower reconstruction quality than sampling wavelet-type angular basis functions on a regular grid. Again, this result may be counter-intuitive because most compressive sensing algorithms perform best when random measurement matrices are used. However, these usually assume a dense random matrix $\mathbf{\Phi}$ (simulated in Fig. 3.4), which is not physically realizable in an ASP sensor. One may believe that a randomization of the available degrees of freedom of the measurement system may be a good approximation of the fully random matrix, but this is clearly not the case. We have not experimented with optical layouts that are optimized for a particular dictionary [135], but expect such codes to further increase reconstruction quality.

3.5 Implementation

3.5.1 Angle Sensitive Pixel Hardware

A prototype ASP light field camera was built using an angle sensitive pixel array sensor [190]. The sensor consists of 24 different ASP types, each of which has a unique response to incident angle. Since a single pixel generates a pair of outputs, a total of 48

distinct angular measurements are read out from the array. Recall from Section 3.3 that ASP responses are characterized by the parameters α, β and γ which define the phase and two dimensional angular frequency of the ASP. The design includes three groups of ASPs that cover low, medium, and high frequencies with β values of 12, 18 and 24, respectively. The low and high frequency groups of ASPs have orientations (γ in degrees) of 0, 90 and $\pm$45 whereas the mid frequency group is staggered in frequency space with respect to the other two and has γ values of $\pm$22.5 and $\pm$67.5. Individual ASPs are organized into a rectangular unit cell that is repeated to form the array. Within each tile, the various pixel types are distributed randomly so that any patch of pixels has a uniform mix of orientations and frequencies as illustrated in Figure 3.6. The die size is 5×5mm which accommodates a 96×64 grid of these tiles, or 384×384 pixels in total.

In addition to the sensor chip, the only optical component in the camera is the focusing lens. We used a commercial 50 mm Nikon manual focus lens at an aperture setting of f/1.2. The setup consisting of the data acquisition boards that host the imager chip as well as the lens can be seen in Figure A.1. The target imaging area was staged at a distance of 1m from the sensor which provided a 10:1 magnification. Calibration of the sensor response was performed by imaging a tiny (2mm diameter), back-illuminated hole positioned far away from the focal plane. Figure 3.6 shows the captured angular point spread function for all 24 ASP types. These responses were empirically fitted and resampled to form the rows of the projection matrix $\mathbf{\Phi}$ for both the linear and nonlinear reconstructions on captured data.

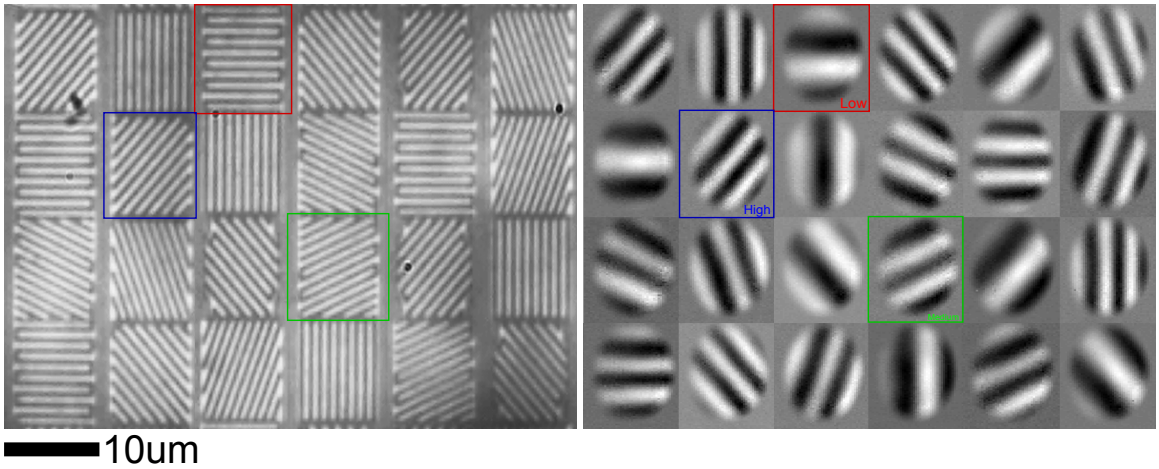


Figure 3.6: Microscopic image of a single 6×4 pixel tile of the ASP sensor (left). We also show captured angular point spread functions (PSFs) of each ASP pixel type (right).

3.5.2 Software Implementation

The compressive part of our software pipline closely follows that of Marwah et al. [135]. Conceptually, nonlinear reconstructions depend on an offline dictionary learning phase, followed by an online reconstruction over captured data. To avoid the challenges of large-scale data collection with our prototype hardware, we used the dictionaries provided by Marwah et al. to reconstruct light fields from the prototype hardware. Dictionaries used to evaluate depth of field in Figure 3.4 were learned using KSVD [4].

Online reconstruction was implemented by the Alternating Direction Method of Multipliers (ADMM) [19] with paramaters $\lambda = 10^{-5}$, $\rho = 1$, and $\alpha = 1$, to solve the ℓ_1-regularized regression (BPDN) of Equation 3.7. RAW images captured by the ASP sensor were subdivided into sliding windows with 9×9 pixels; small 4D light field patches were reconstructed for each of the windows, each one with 5×5 angles. The sliding reconstruction window was translated in one pixel increments over the full 384×384 pixel sensor image and the results were integrated with an average filter. Reconstructions were com-

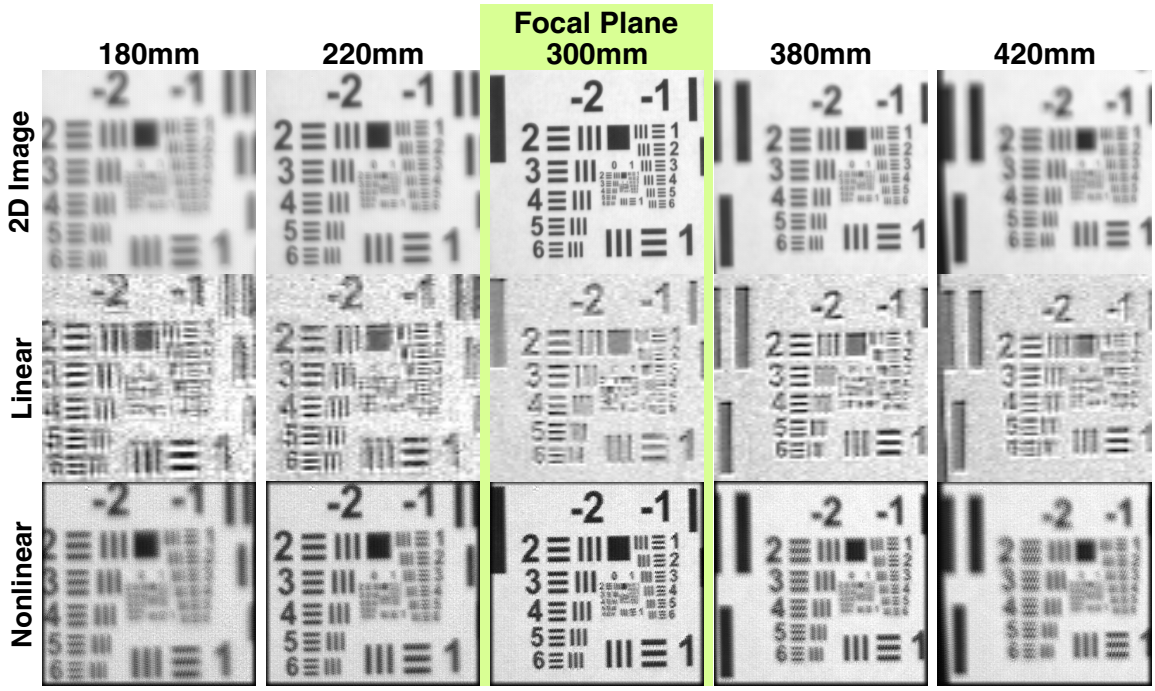


Figure 3.7: Evaluation of prototype resolution. We capture images of a resolution target at different depths and compare 2D image (top), center view of the linearly reconstructed light field (center), and center view of the nonlinearly reconstructed light field (bottom).

puted on an 8-core Intel Xeon workstation with 16GB of RAM. Average reconstruction time for each of the experiments in Section 3.6 was approximately 8 hours. In contrast, linear reconstruction algorithms are significantly faster and take less than one minute for each result.

3.6 Results

This section shows an overview of experiments with the prototype camera.

In Figure 3.7, we evaluate the resolution of the device for all three proposed reconstruction algorithms. As expected for a conventional 2D image, the depth of field is limited by the f-number of the imaging lens, resulting in out-of-focus blur for a resolution chart that moves away from the focal plane (top row). The proposed linear reconstruction recovers the 4D light field at a low resolution (center row). Due to the lack of an optical anti-aliasing filter in the camera, aliasing is observed in the reconstructions. The anti-aliasing filter would remove these artifacts but also decrease image resolution. The resolution of the light field recovered using the sparsity-constrained nonlinear methods has a resolution comparable to the in-focus 2D image. Slight artifacts in the recovered resolution charts correspond to those observed in noise-free simulations (cf. Fig 3.5). We believe these artifacts are due to the large compression ration—25 light field views are recovered from a single sensor image via sparsity-constrained optimization.

We show additional comparisons of the three reconstruction methods for a more complex scene in Figure 3.8.

Figure 3.9 shows several scenes that we captured in addition to those already shown

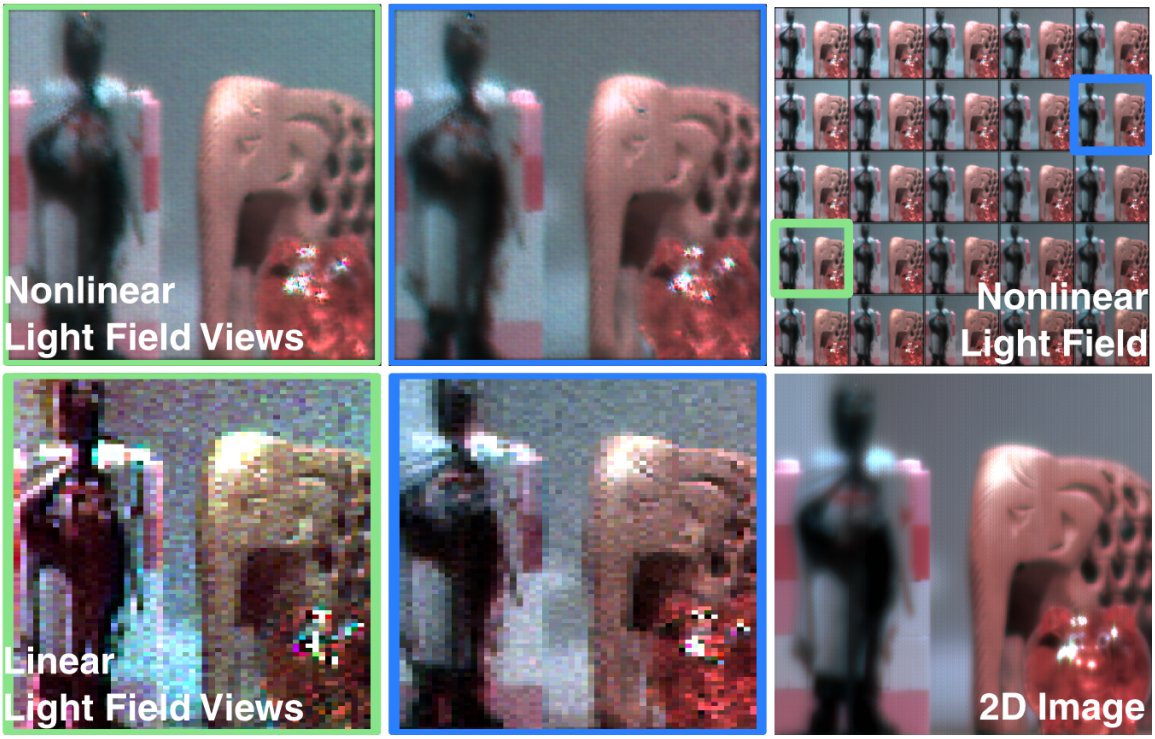


Figure 3.8: Comparison of different reconstruction techniques for *the same captured data*. We show reconstruction of a 2D image (bottom right), a low-resolution light field via linear reconstruction (bottom left and center), and a high-resolution light field via sparsity-constrained optimization with overcomplete dictionaries (top). Whereas linear reconstruction trades angular for spatial resolution—thereby decreasing image fidelity—nonlinear reconstructions can achieve an image quality that is comparable to a conventional, in-focus 2D image for each of 25 recovered views.

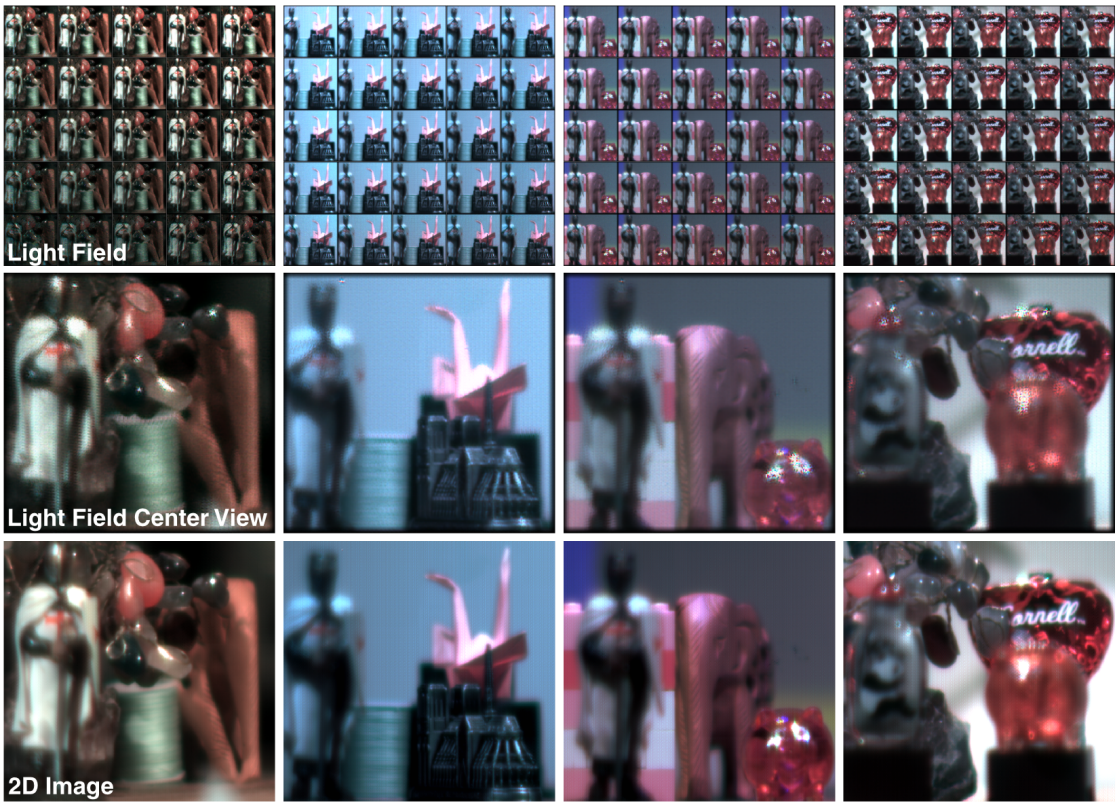


Figure 3.9: Overview of captured scenes showing mosaics of light fields reconstructed via sparsity-constrained optimization (top), a single view of these light fields (center), and corresponding 2D images (bottom). These scenes exhibit a variety of effects, including occlusion, refraction, specularity, and translucency. The resolution of each of the 25 light field views is similar to that of the conventional 2D images.

in Figures A.1 and 3.8. Animations of the recovered light fields for all scenes can be found in the supplementary video. We deliberately include a variety of effects in these scenes that are not easily captured in alternatives to light field imaging (e.g., focal stacks or range imaging), including occlusion, refraction, and translucency. Specular highlights, as for instance seen on the glass piglet in the two scenes on the right, often lead to sensor saturation, which causes artifacts in the reconstructions. This is a limitation of the proposed reconstruction algorithms.

Finally, we show in Figure 3.10 that the recovered light fields contain enough parallax to

Figure 3.10: Refocus of the "Knight & Crane" scene.

allow for post-capture image refocus. Chromatic aberrations in the recorded sensor image and a limited depth of field of each recovered light field view place an upper limit on the resolvable resolution of the knight (right).

3.7 Compressive Light Field Reconstructions using Deep Learning

One major limitation of the previous sections is the computational time necessary to perform the nonlinear reconstruction, partly due to the iterative solvers to solve the ℓ_1 minimization problem. In this section, we present a deep learning approach using a new, two branch network architecture consisting jointly of an autoencoder and a 4D CNN to recover a high resolution 4D light field from a single coded 2D image. This network achieves average PSNR values of 26-28 dB on a variety of light fields, and outperforms existing state-of-the-art baselines such as generative adverserial networks and dictionary-based learning. In addition, reconstruction time is decreased from 35 minutes to 6.7 minutes as compared

to the dictionary method for equivalent visual quality. These reconstructions are performed at small sampling/compression ratios as low as 8%, which allows for cheaper coded light field cameras. We test our network reconstructions on synthetic light fields, simulated coded measurements of a dataset of real light fields captured from a Lytro Illum camera, and real ASP data from the previous sections in this chapter. The combination of compressive light field capture with deep learning allows the potential for real-time light field video systems in the future.

3.7.1 Related Work

Light Field Reconstruction: Several techniques have been proposed to increase the spatial and angular resolution of captured light fields. These include using explicit signal processing priors [118] and frequency domain methods [166]. The work closest to our own is the introduction of compressive light field photography [135] that uses learned dictionaries to reconstruct light fields, and extending that technique to Angle Sensitive Pixels [84] which was described in the previous chapter. We replace that framework by using deep learning to perform both the feature extraction and reconstruction with a neural network. Similar to our work, researchers have recently used deep learning networks for view synthesis [96] and spatio-angular superresolution [214]. However, all these methods start from existing 4D light fields, and thus they do not recover light fields from compressed or multiplexed measurements.

Compressive Sensing: There have been numerous works in the areas of compressed sensing [27] and algorithms to recover the original signal. The classical algorithms [41, 26, 25] rely on the assumption that the signal to be reconstructed in transform domains like wavelet, DCT or a data dependent pre-trained dictionary. More sophisticated

algorithms include model-based methods [10, 102] and message-passing algorithms [42] which impose a complex image model to perform reconstruction. However, all of these algorithms are iterative and hence are not conducive for fast reconstruction. Similar to our work, deep learning has been used for recovering 2D images from compressive measurements at faster speeds than iterative solvers [107, 138], and even for compressive video [86]. [138, 86] proposed stacked-denoising autoencoders to perform CS image and video reconstruction respectively. [107] show that CNNs which are traditionally used for inference tasks which demand spatial invariance, e.g image recognition, can also be used for CS image reconstruction. We marry the benefits of the two types of architectures mentioned above and propose a novel architecture to 4D light fields which introduce additional challenges and opportunities for deep learning + compressive sensing.

3.7.2 Deep Learning for Light Field Reconstruction

We first discuss the datasets of light fields we use for simulating coded light field capture along with our training strategy before discussing our network architecture.

Light Field Simulation and Training

One of the main difficulties for using deep learning for light field reconstructions is the scarcity of available data for training, and the difficulty of getting ground truth (especially for compressive light field measurements). We employ a mixture of simulation and real data to overcome these challenges in our framework (see Figure 3.11 for representative light fields we used).

Synthetic Light Field Archive: We use synthetic light fields from the Synthetic Light

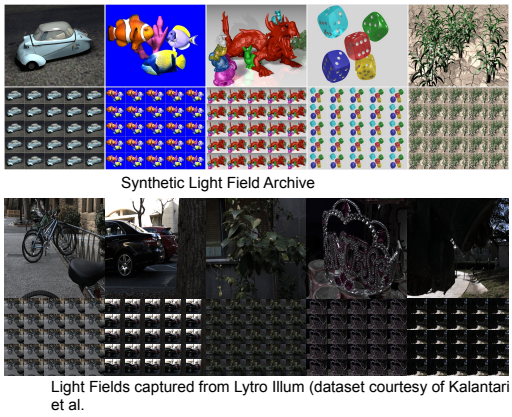


Figure 3.11: Left: Shown are the training sets, both synthetic and the UCSD dataset. Right: The diagram shows an overview of our method to train the network for light-field reconstruction.

Field Archive [203] which have resolution $(x, y, \theta, \phi) = (593, 840, 5, 5)$. Since the number of parameters for our fully-connected layers would be prohibitively large with the full light field, we split the light fields into $(9, 9, 5, 5)$ patches and reconstruct each local patch. We then stitch the light field back together using overlapping patches to minimize edge effects.

Our training procedure is as follows. We pick 50,000 random patches from four synthetic light fields, and simulate coded capture by multiplying by Φ to form images. We then train the network on these images with the labels being the true light field patches. Our training/validation split was 85:15. We finally test our network on a brand new light field never seen before, and report the PSNR as well as visually inspect the quality of the data. In particular, we want to recover parallax in the scenes, i.e. the depth-dependent shift in pixels away from the focal plane as the angular view changes.

Lytro Illum Light Field Dataset: In addition to synthetic light fields, we utilize real light field captured from a Lytro Illum camera [96]. To simulate coded capture, we use the same Φ models for each type of camera and forward model the image capture process, resulting in simulated images that resemble what the cameras would output if they captured

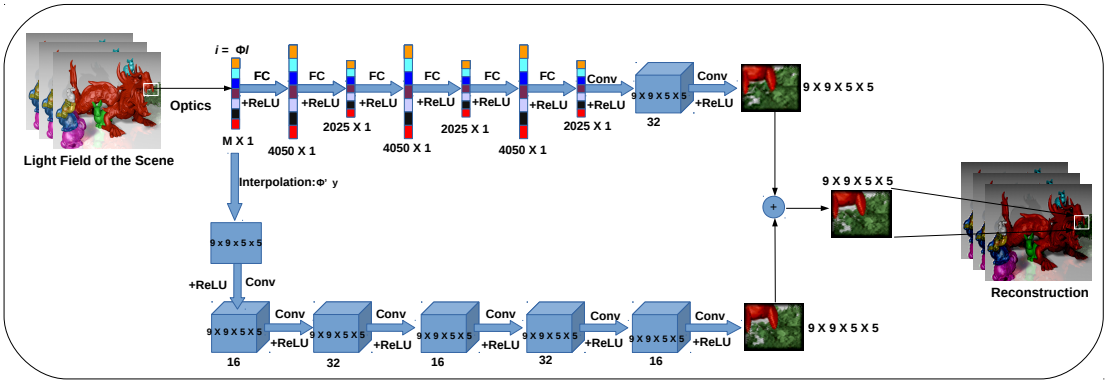


Figure 3.12: The figure shows the 2-branch architecture for light-field reconstruction. Measurements for every patch of size 5 X 5 X 9 X 9 are fed into two streams, one consisting of 6 fully connected and 2 4D convolution layers, and the other consisting of 5 4D convolutional layers. The outputs of the two branches are added with equal weights to obtain the final reconstruction for the patch. Note that the size of filters in all convolution layers is 3 X 3 X 3 X 3.

that light field. There are a total of 100 light fields, each of size $(364, 540, 14, 14)$. For our simulation purposes, we use only views $[6, 10]$ in both θ and ϕ, to generate $5x5$ angular viewpoints. We extract 500,000 patches from these light fields of size 9x9x5x5, simulate coded capture, and use 85:15 training/validation split.

Network Architecture

Our network architecture consists of a two branch network, which one can see in Figure 3.12. In the upper branch, the 2D input image is vectorized to one dimension, then fed to a series of fully connected layers that form an autoencoder (i.e. alternating contracting and expanding layers). This is followed by one 4D convolutional layer. The lower branch (a 4D CNN) uses a fixed interpolation step of multiplying the input image by Φ^T to recover a 4D spatio-angular volume, and then fed through a series of 4D convolutional layers with ReLU, a pointwise rectified linear function which is 0 for negative numbers and $y = x$ else.

Finally the outputs of the two branches are combined with weights of 0.5 to estimate the light field.

Autoencoders are useful at extracting meaningful information by compressing inputs to hidden states [189], and our autoencoder branch helped to extract parallax (angular views) in the light field. In contrast, our 4D CNN branch utilizes information from the linear reconstruction by interpolating with Φ^T and then cleaning the result with a series of 4D convolutional layers for improved spatial resolution. Combining the two branches thus gave us good angular recovery along with high spatial resolution. In Figure 3.13, we show the results of using solely the upper or lower branch of the network versus our two stream architecture. The two branch network gives a 1-2 dB PSNR improvement as compared to the autoencoder or 4D CNN alone, and one can observe the sharper detail in the inlets of the figure.

For the loss function, we observed that the regular ℓ_2 loss function gives decent reconstructions, but the amount of parallax and spatial quality recovered in the network at the extreme angular viewpoints were lacking. To remedy this, we employ the following weighted ℓ_2 loss function:

$$L(l, \hat{l}) = \sum_{\theta,\phi} W(\theta, \phi) \cdot ||l(x, y, \theta, \phi) - \hat{l}(x, y, \theta, \phi)||_2^2, \tag{3.8}$$

where $W(\theta, \phi)$ are weights that increase for higher values of θ, ϕ[1]. This biases the network to reconstruct the extreme angular views with higher fidelity.

[1] The weight values were picked heuristically for large weights away from the center viewpoint: $W(\theta, \phi) =$

$$\begin{pmatrix} \sqrt{5} & 2 & \sqrt{3} & 2 & \sqrt{5} \\ 2 & \sqrt{3} & \sqrt{2} & \sqrt{3} & 2 \\ \sqrt{3} & \sqrt{2} & 1 & \sqrt{2} & \sqrt{3} \\ 2 & \sqrt{3} & \sqrt{2} & \sqrt{3} & 2 \\ \sqrt{5} & 2 & \sqrt{3} & 2 & \sqrt{5} \end{pmatrix}$$

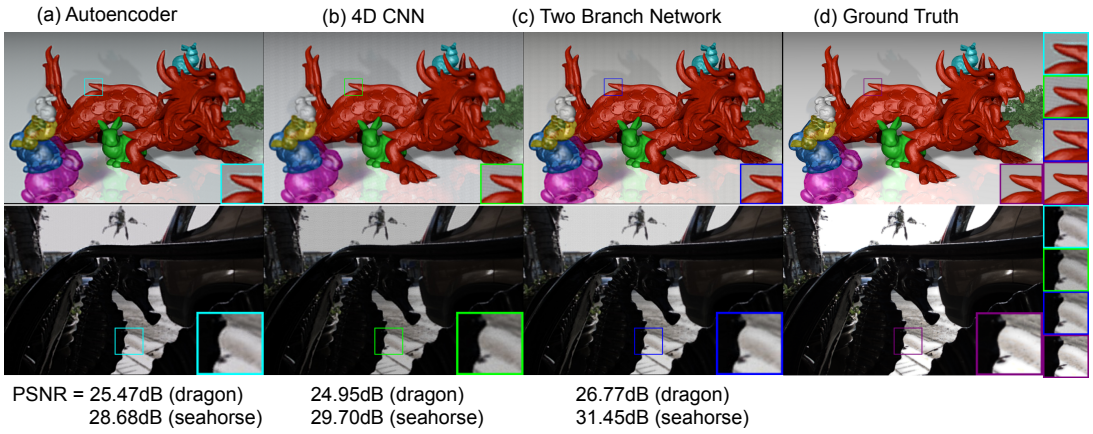


Figure 3.13: The figure shows the reconstruction for two scenes, dragons and seahorse. For both the scenes, we obtain better results in terms of PSNR for the two-stream network than the two individual branches, autoencoder and 4D CNN. This corroborates the need for two-branch network, which marries the benefits of the two branches.

Training Details

All of our networks were trained using Caffe [92] and using a NVIDIA Titan X GPU. Learning rates were set to $\lambda = .00001$, we used the ADAM solver [103], which is a type of stochastic gradient descent with adaptive momentum, and models were trained for about 60 epochs for 7 hours. We also finetuned models trained on different Φ matrices, so that switching the structure of a Φ matrix did not require training from scratch, but only an additional few hours of finetuning.

For training, we found the best performance was achieved when we trained each branch separately on the data, and then combined the branches and finetuned the model further on the data. Training from scratch the entire two branch network led to suboptimal performance of 2-3 dB in PSNR, most likely because of local minima in the loss function.

GANs

We tested our network architecture against a state-of-the-art baseline: generative adverserial networks (GANs) [64] which have been shown to work well in image generation and synthesis problems [148, 115]. In classical GAN formulations, an uniform random vector z is mapped to an image, and two networks, a generator G and discriminator D are alternatively trained according to a min-max game-theoretic optimization. The goal is for the generator G to generate an image $G(z)$ that fools discriminator D.

We use the autoencoder branch as the generator network, and build a discriminator network consisting of 4D convolutions followed by a fully connected layer aimed to decide if a given light field reconstruction was ground truth or a network output. In our case, instead of inputting a random vector, we input the coded measurements i to the generator network. The discriminator network determines if a reconstructed light field is fake or not. One can interpret the discriminator network as classifying the light-field reconstruction from generator network as having parallax or not. While this method has shown considerable improvements on 2D image problems, we found GANs to perform slightly worse than our two branch network by about 2 dB in PSNR, as you can see in Figure 3.14. The GAN outputted lower resolution scenes with noticeable color and edge artifacts as one can see in the inlets.

3.7.3 Experimental Results

In this section, we show experimental results on both simulated light fields, real light fields with simulated capture, and finally real data taken from a prototype ASP camera [84]. We compare both visual quality and reconstruction time for our reconstructions, and com-

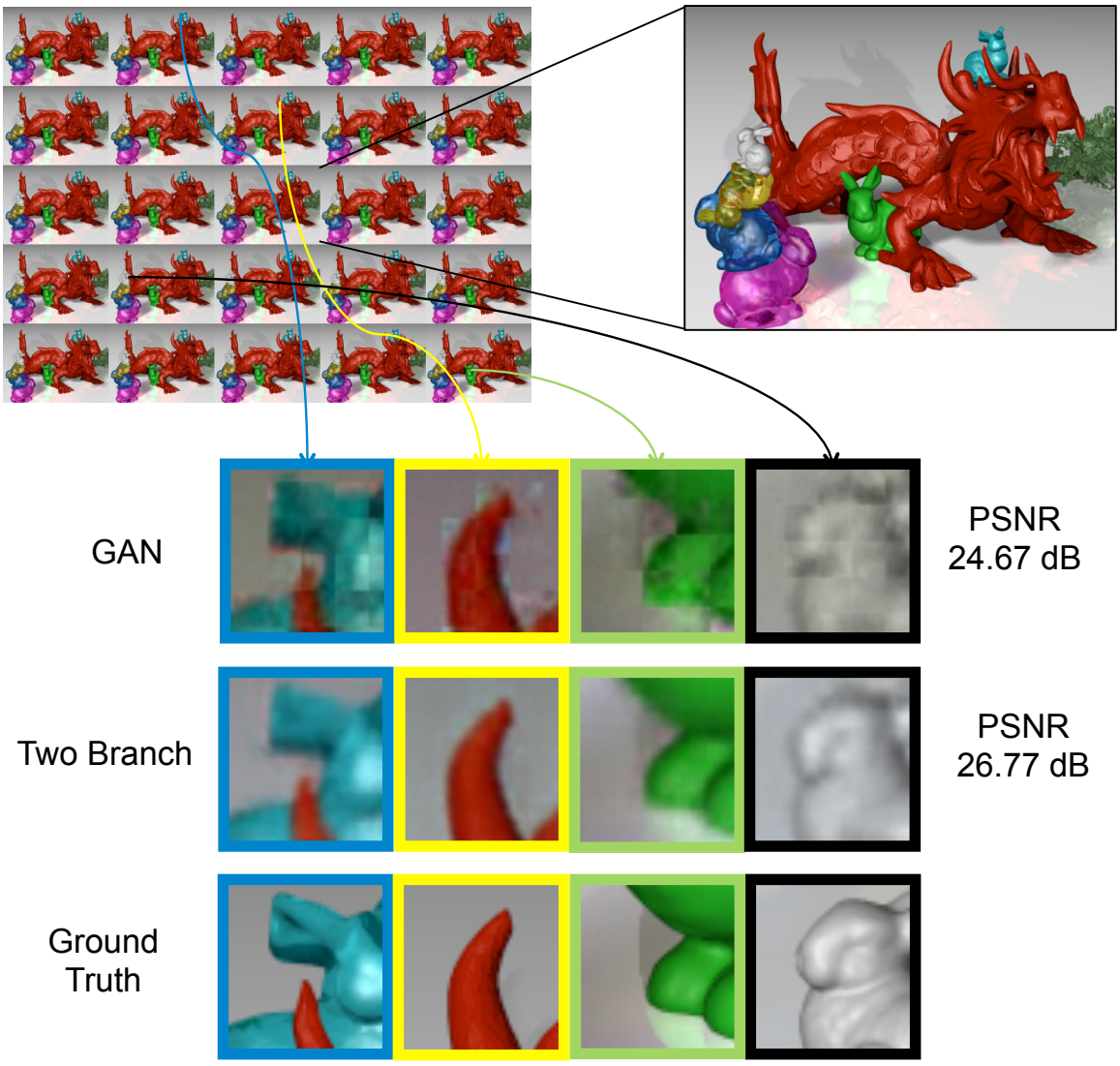


Figure 3.14: The figure compares the reconstructions for the dragons scene using the two-branch network proposed as well as the autoencoder in conjunction with GAN. Clearly, the quality of the reconstructions for the former (26.77 dB) is greater than that for the latter (24.67 dB).

pare against baselines for each dataset.

Synthetic Experiments

We first show simulation results on the Synthetic Light Field Archive. We used as our baseline the dictionary-based method from [135, 84] with the dictionary trained on synthetic light fields, and we use the dragon scene as our test case. We utilize three types of Φ matrices, a random Φ matrix that represents the ideal 4D random projections matrix (satisfying the restricted isometry property or RIP [24]), but is not physically realizable in hardware (rays are arbitrarily summed from different parts of the image sensor array). We also use Φ for coded masks placed in the body of the light field camera, a repeated random code that is periodically shifted in angle across the array. Finally, we use the Φ matrix for ASPs which consists of 2D oriented sinusoidal responses to angle. As you can see in Figure 3.15, the ASPs and the Mask reconstructions perform slightly better than the ideal random projections, perhaps since the compression ratio is low at 8%, which might not satisfy the limits of the compressed sensing theory. The reconstructions do suffer from blurred details in the zoomed inlets, which means that there is still spatial resolution that is not recovered by the network.

Compression ratio is the ratio of independent coded light field measurements to angular samples to reconstruct in the light field for each pixel. This directly corresponds to the number of rows in the Φ matrix which correspond to one pixel. We show a sweep of the compression ratio and measure the PSNR for both the mask and ASP light field cameras, as you can see in Figure 3.16. The PSNR degrades gracefully as the number of measurements becomes smaller, with ASPs and coded mask cameras still achieving above 25 dB at a compression ratio of 8%.

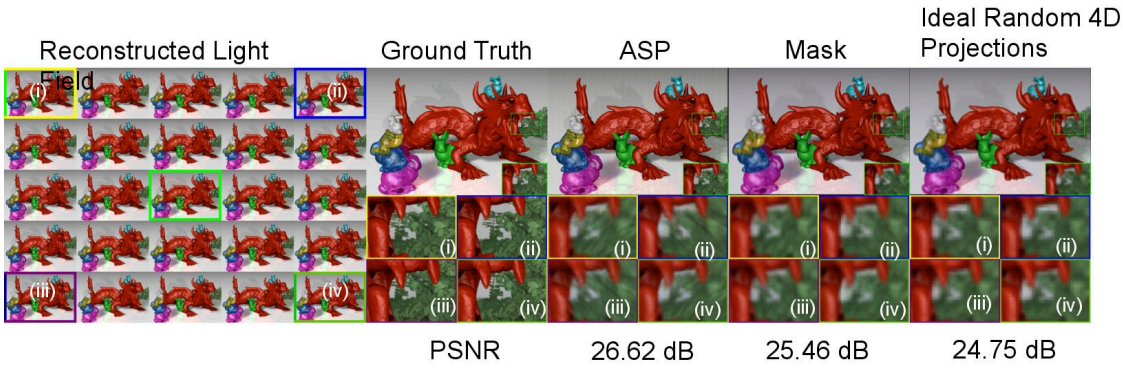


Figure 3.15: The figure compares the reconstructions for the dragons scene for different encoding schemes, ASP, Mask and Ideal Random 4D projections (CS) using the two-branch network. We obtain better quality reconstruction results for ASP ()26.62 dB) as compared to Mask (25.46 dB) and CS (24.75 dB).

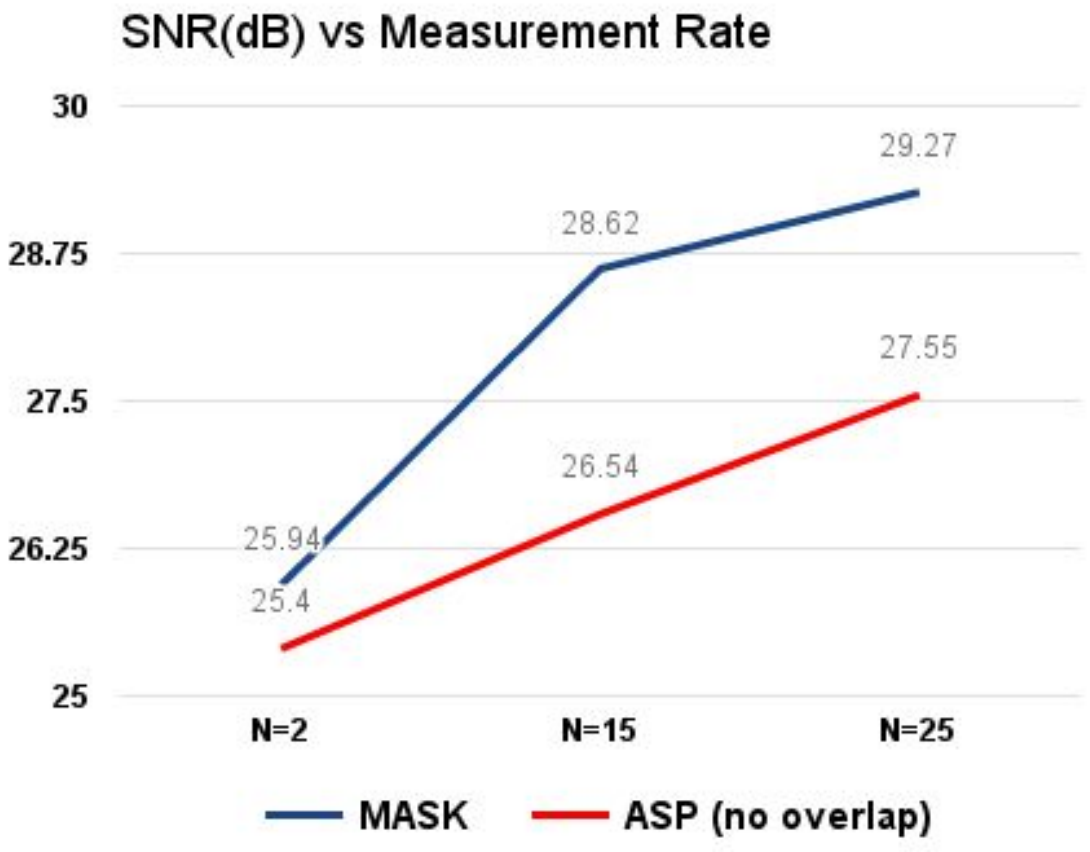


Figure 3.16: The figure shows the variation of PSNR of reconstructions with the number of measurements for dragons scene for ASP and Mask using the two-stream network.

Metrics	Noiseless	Std 0.1	Std 0.2
PSNR (Ours)	26.77	26.74	26.66
PSNR (Dictionary)	25.80	21.98	17.40
Time (Ours)	242	242	242
Time (Dictionary)	3786	9540	20549

Table 3.2: The table shows how PSNR varies for different levels of additive Gaussian noise. It is clear that our method is extremely robust to high levels of noise and provides high PSNR reconstructions, while for the dictionary method, the quality of the reconstructions dip sharply with noise. Also shown is the time taken to perform the reconstruction. For our method, the time taken is only 242 s whereas for dictionary learning method, it can vary from 1 hour to nearly 7 hours.

Noise: We also tested the robustness of the networks to additive noise in the input images. We simulated Gaussian noise of standard deviation 0.2 and 0.4, and recorded both the PSNR and reconstruction time which is displayed in Table 3.2. Note that the dictionary-based algorithm takes longer to process noisy patches due to its iterative ℓ_1 solver, while our network has the same flat run time regardless fo the noise level. This is a distinct advantage of neural network-based methods over the iterative solvers. The network also seems resilient to noise in general, as our PSNR remained about 26 dB.

Lytro Illum Light Fields Dataset: We show our results on the UCSD dataset in Figure 3.17. As a baseline, we compare against the method from Kalantari et al. [96] which utilize 4 input views from the light field and generate the missing angular viewpoints with a neural network. Our network model achieves higher PSNR values of 28-29 dB on these real light fields. While Kalantari et al. method achieves PSNR $\geq$ 30dB on this dataset, this is starting from a 4D light field captured by the Lytro camera and thus does not have to uncompress coded measurements. Their effective compression ratio is 16% without any encoding, while our network is processing compression ratios of 8% with encoding. Our method is also slightly faster as their network takes 147 seconds to reconstruct the full light field, while our method reconstructs a light field in 80 seconds (both on a Titan X GPU).

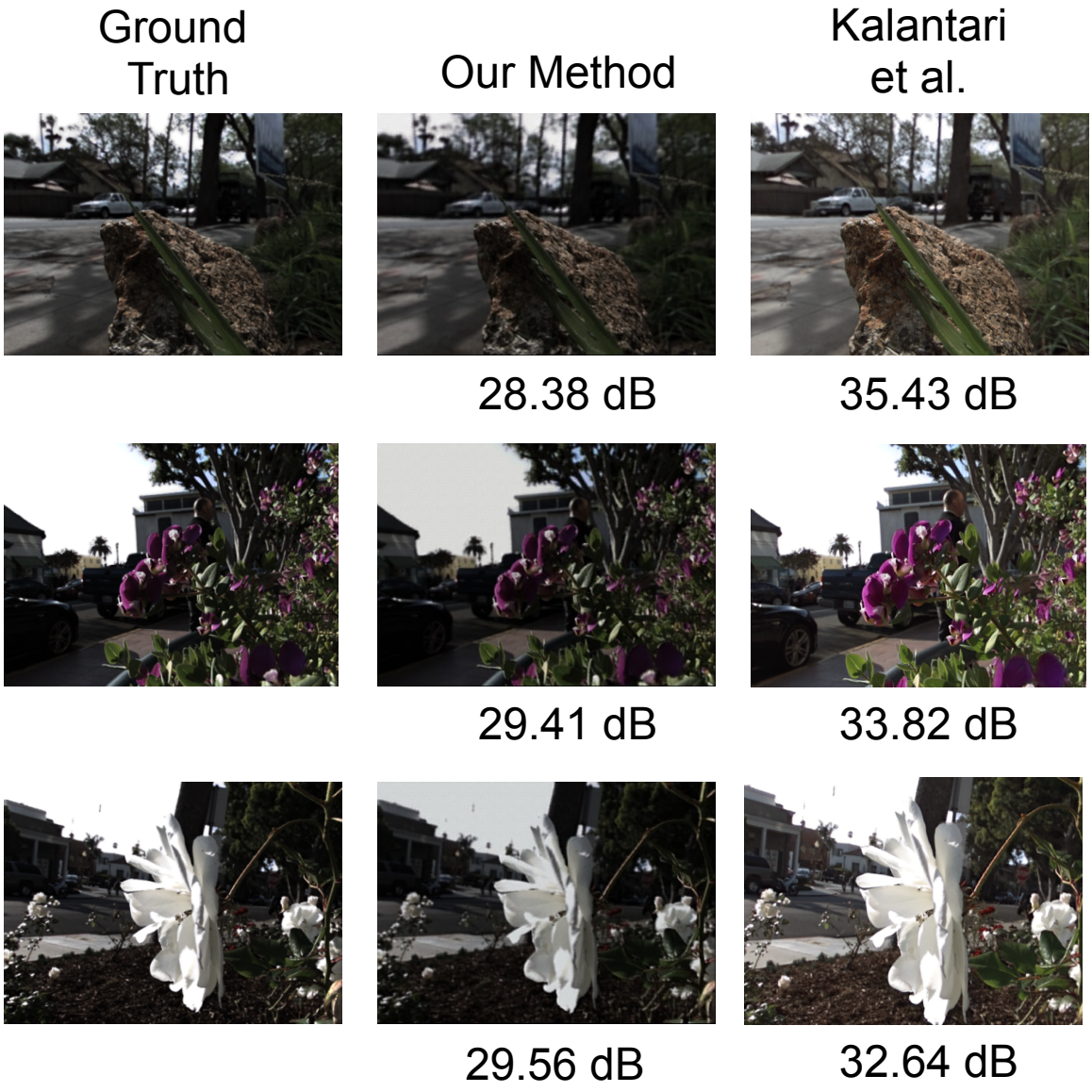


Figure 3.17: UCSD Dataset reconstruction comparison. SNR is computed only for the central 5x5 views

Real Experiments

Finally, to show the feasibility of our method on a real compressive light field camera, we use data collected from a prototype ASP camera [84], and that was described in earlier sections in this chapter. This data was collected on an indoors scene, and utilized three color filters to capture color light fields of size (384,384,3).

Since we don't have training data for these scenes, we train our two branch network on synthetic data, and then apply a linear scaling factor to ensure the testing data has the same statistics as the training data. We also change our Φ matrix to match the actual sensors response and measure the angular variation in our synthetic light fields to what we expect from the real light field. See Figure 3.18 for our reconstructions. We compare our reconstructions against the dictionary-based method. For all reconstruction techniques, we apply post-processing filtering to the image to remove periodic artifacts due to the patch-based processing and non-uniformities in the ASP tile, as done in [84].

We first show the effects of stride, defined as the number of pixels for which the patch shifts to extract measurements from the image, for overlapping patch reconstructions for the light fields, as shown in Figure 3.19. Our network model takes a longer time to process smaller stride, but improves the visual quality of the results. This is a useful tradeoff between visual quality of results and reconstruction time in general.

As you can see, the visual quality of the reconstructed scenes from the network are on-par with the dictionary-based method, but with an order of magnitude faster reconstruction times. A full color light field with stride of 5 in overlapping patches can be reconstructed in 40 seconds, while an improved stride of 2 in overlapping patches yields higher quality reconstructions for 6.7 minutes of reconstruction time. The dictionary-based method in

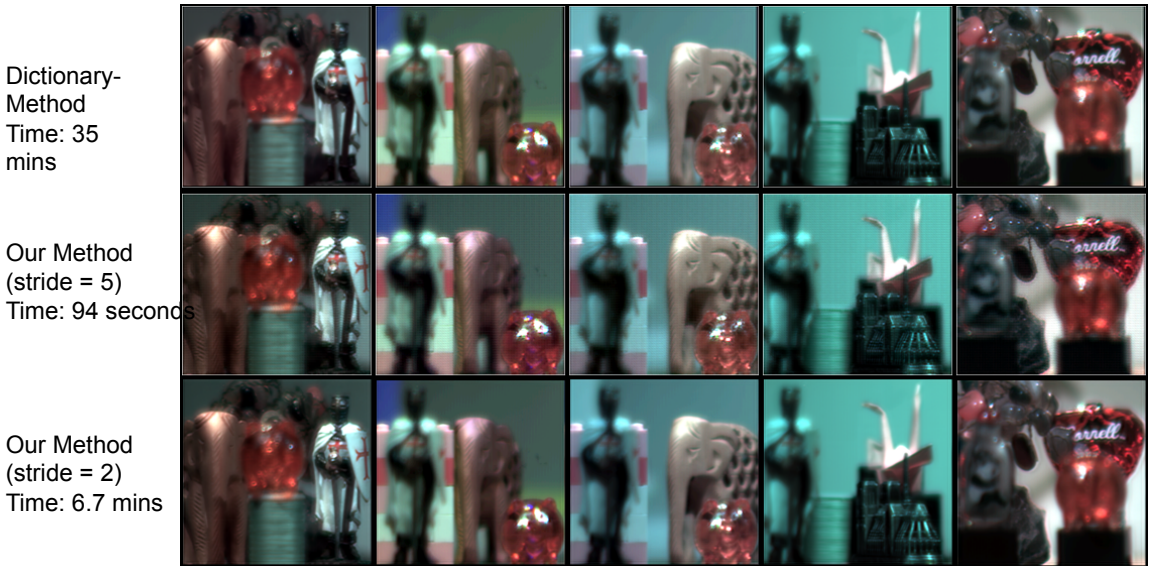


Figure 3.18: The figure shows the reconstructions for the real data from the ASP measurements using our method (for stride 5 and stride 2) and dictionary method (for stride 5), and the corresponding time taken. It is clear that the quality of the reconstructions for our method is comparable as that using the dictionary learning method, although the time taken for our method (94 seconds) is an order less than that for the dictionary learning method (35 minutes).

contrast takes 35 minutes for a stride of 5 to process these light fields. The recovered light fields have parallax, although some distortions exist that may be due to optical abberations, a mismatch between the real Φ response and the model Φ, and higher noise in the real data as compared to synthetic data. However, we believe these results represent the potential for using neural networks to recover 4D light fields from real coded light field cameras.

3.7.4 Discussion

In this section, we have presented a new network architecture to recover 4D light fields from compressive measurements via coded light field cameras. The two branch structure of a traditional autoencoder and a 4D CNN lead to superior performance, outperforming other network topologies such as GANs. We benchmark our results on both synthetic and real

Figure 3.19: Comparison of non-overlapping patches and overlapping patches with strides of 11 (non-overlapping), 5, and 2.

light fields, and achieve good visual quality while reducing reconstruction time to minutes compared to the dictionary-based method.

3.7.5 Limitations

There are some limitations to our work. Since acquiring ground truth for coded light field cameras is difficult, there is no possibility of fine tuning our model for improved performance. In addition, it is hard to determine exactly the Φ matrix without careful optical calibration, and this response is dependent on the lens and aperture settings during capture time. All of this information is hard to feed into a neural network to adaptively learn, and leads to a mismatch between the statistics of training and testing data.

3.7.6 Future Directions

One future direction is to have the network jointly learn optimal codes for capturing light fields with the reconstruction technique, similar to the work by Chakrabarti [30] and Mousavi et al. [138], to help design new types of coded light field cameras. While our work has focused on processing single frames of light field video efficiently, we could explore performing coding both in the spatio-angular domain as well as in the temporal domain. This would help improve the compression ratio for these sensors, and potentially lead to light field video that is captured at interactive (0.1-15 FPS) frame rates. Finally, it would be interesting to perform inference on compressed light field measurements directly (similar to the work for inference on 2D compressed images [131, 108]) that aims to extract meaningful semantic information. All of these future directions point to a convergence between compressive sensing, deep learning, and computational cameras for enhanced light field imaging.

CHAPTER 4

POLARIZATION

While ASP diffraction gratings were primarily designed for angular sensitivity, this chapter[1] observes that the gratings also can sense the polarization of incoming light. This yields an exciting additional plenoptic dimension that can be used for computer vision. We characterize the polarization effect of ASPs which is added to our forward model for plenoptic capture from the previous chapter. Finally, we show two applications: imaging stress-induced birefringence and removing specular highlights from a light field depth map.

4.1 Polarization Response

Recall that ASP's forward imaging model can be written as follows:

$$I = A(\theta) \cdot (1 + m\cos(\beta(\cos(\gamma)\theta_x + \sin(\gamma)\theta_y) + \alpha)). \tag{4.1}$$

We note that the aperture function $A(\theta)$ is a Gaussian function that represents extreme angles being attenuated by the pixel aperture. It is this aperture function that we will extend to incorporate polarization since polarization is a common mode effect in these differential ASP pixels.

To extend the forward imaging model to include polarization, we measured the impulse response of ASPs by imaging a blurred out point source of light placed at optical infinity with a polarizing filter at different polarization angles in front of the camera. The ASP sensor sees variation in incidence angle across the blurred out spot as shown in Fig. 1. By

[1]The work in this chapter was originally presented in S. Jayasuriya et al, "Dual light field and polarization imaging using diffractive CMOS image sensors", Optics Letters 2015 [91].

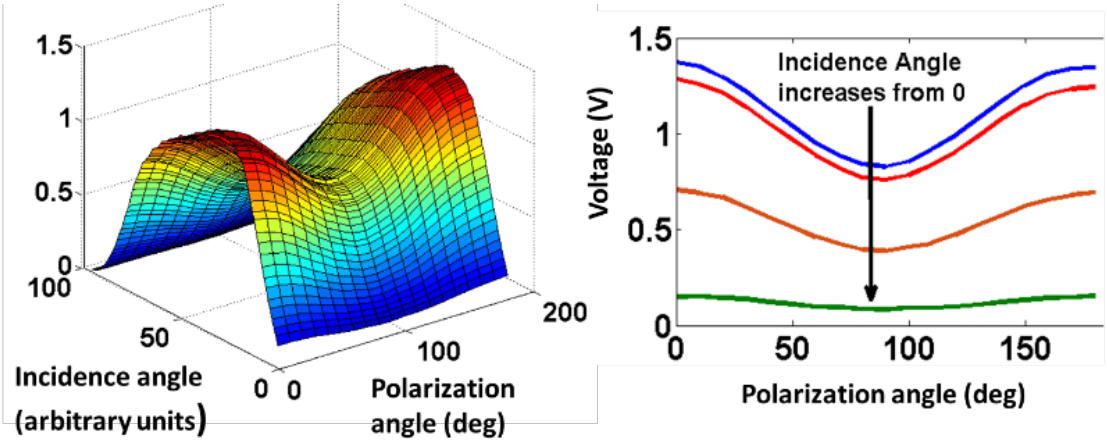


Figure 4.1: Left plot displays polarization angle versus incidence angle in 1D with response $\cos(2\psi - \gamma) \cdot e^{\theta_x}$. Right plot shows cross-section of this response at different incident angles.

adding the two sub-pixels with complementary phase separations ($\alpha = 0, \pi$ or $\frac{\pi}{2}, \frac{3\pi}{2}$), we recover the common mode or average intensity at each pixel that shows a fixed aperture response for all pixel types [196]. Thus any variation in the common mode between two pixels is due to that pixels polarization response. Fig. 2 shows this intensity as a function of polarization has a response of $\cos(2\psi + \gamma + \pi)$ where ψ is polarization angle and γ is the grating orientation. This conforms to the intuition that our diffraction gratings act similar to wire grid polarizers. We can therefore have the aperture function of the ASP response be modulated by polarization as $\cos(2\psi + \gamma + \pi) \cdot A(\theta)$.

We also characterized the polarization response with respect to grating orientation as shown in Figure 4.2. From the 0, 45, 90, and 135 degree grating orientation ASPs, we can calculate the angle of polarization from their common mode responses to be

$$\psi = \tan^{-1}\left(\frac{I_{45} - I_{135}}{I_0 - I_{90}}\right).$$

. We show a plot of the measured angle of polarization in Figure 4.3. The degree of linear polarization was measured to be about 25%, and the average deviation between calculated

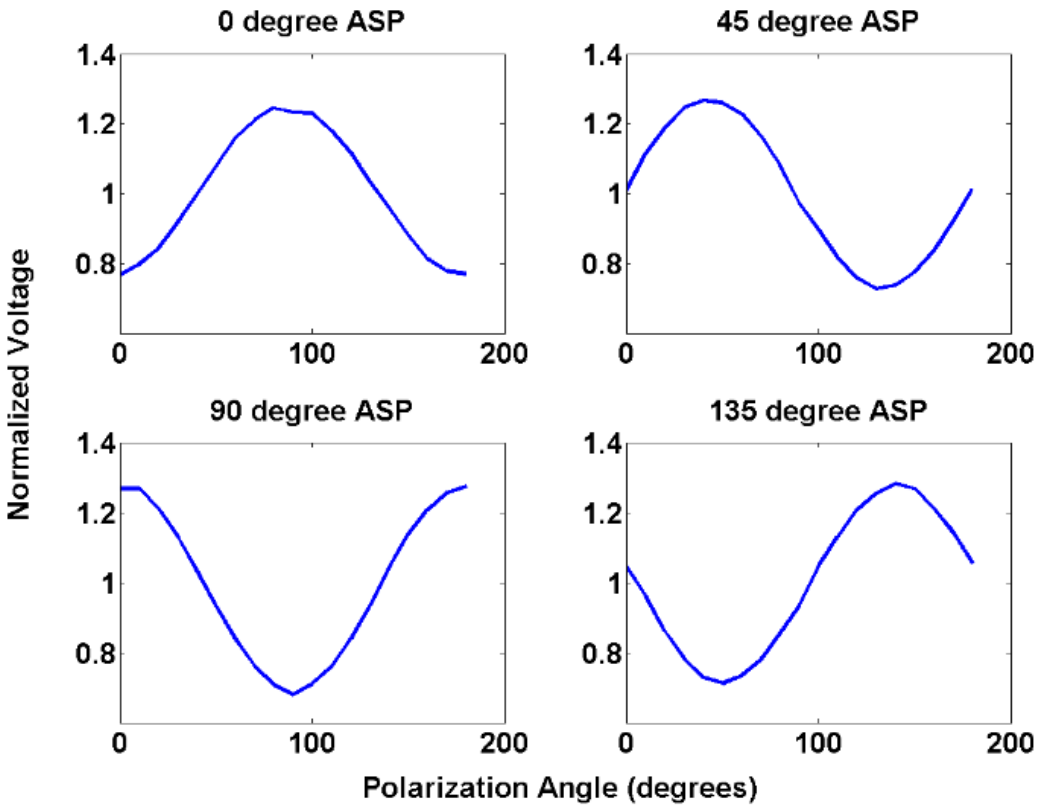


Figure 4.2: Polarization response for ASPs with grating orientation of 0, 45, 90, and 135 degrees. Extinction ratios of approximately 2 were recorded.

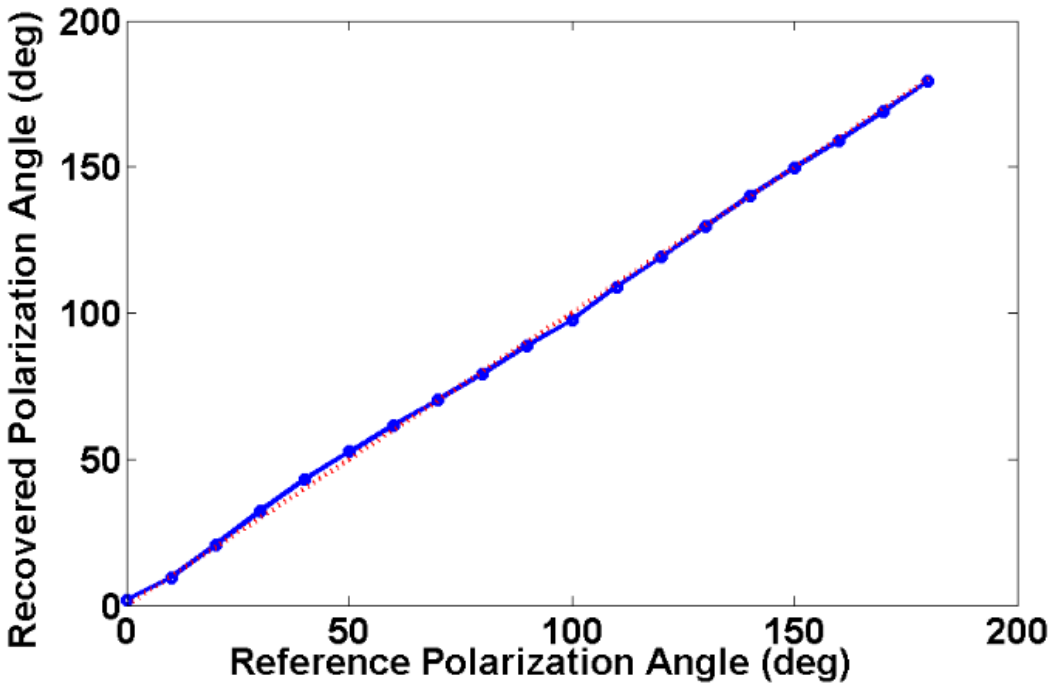


Figure 4.3: Recovered polarization angle versus reference angle. The average deviation was around 1.12 degrees.

and reference angles was 1.12 degrees. The extinction ratio was calculated to be around 2 which is somewhat lower than ratios for other CMOS polarizing sensors with integrated filters (ratio of 6-50 [70, 69, 157]). This is explained by the fact that the pitch of the gratings need to be on the order of the wavelength of light in order to act as diffraction gratings (our pitch was 1μm), and thus prohibits high extinction ratios obtained by wire grid polarizers with sub-wavelength structures [70]. This is a fundamental tradeoff for any grating that wants to simultaneously polarize and diffract visible light.

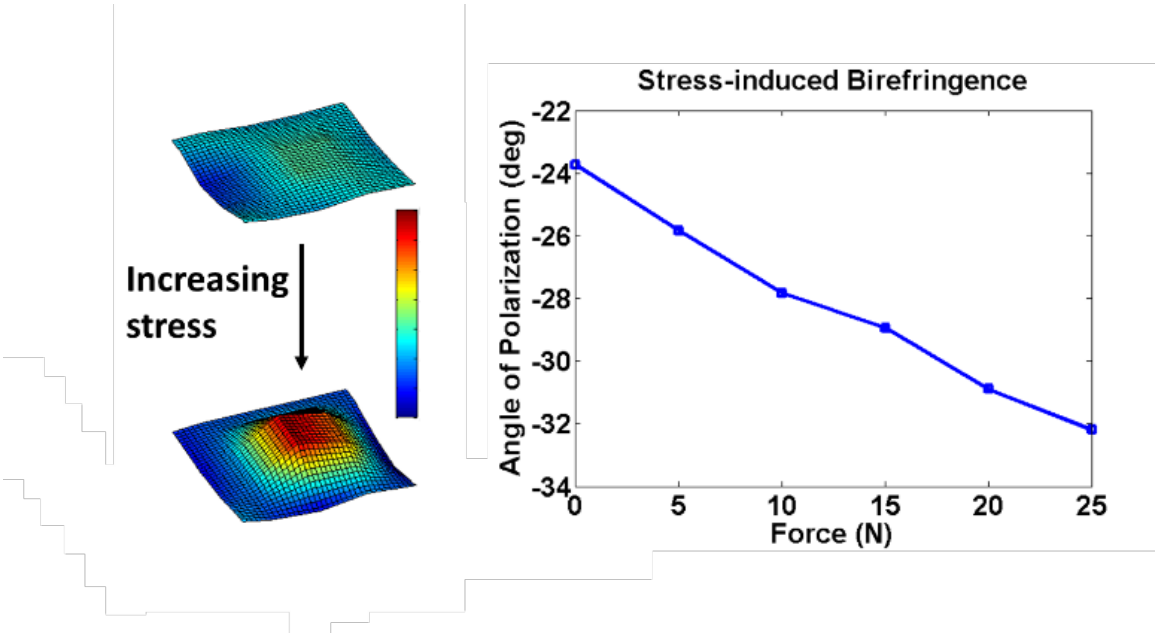


Figure 4.4: Measuring change in polarization angle as a function of applied stress to polyethylene material. The minimum detectable force was 0.5N.

4.2 Applications

In many transparent materials, stress induces birefringence in the material that changes the polarization of light passing through it. Detecting birefringence has several applications in biomedical microscopy, and ASP algorithms such as synthetic aperture refocusing can combine with detecting birefringence to increase the effectiveness of polarization microscopy. In Figure 4.4, we show the results stressing a sheet of clear polyethylene (thickness of 4 mil) with a force gauge while shining polarized light through the sheet. Force applied ranged up to 25 N and reliable minimum detection could be achieved for 0.5 N. Further detection is limited since the signal of interest is around 2-5mV which is close to the noise of the image sensor.

Another important application of polarization imaging is to identify and remove spec-

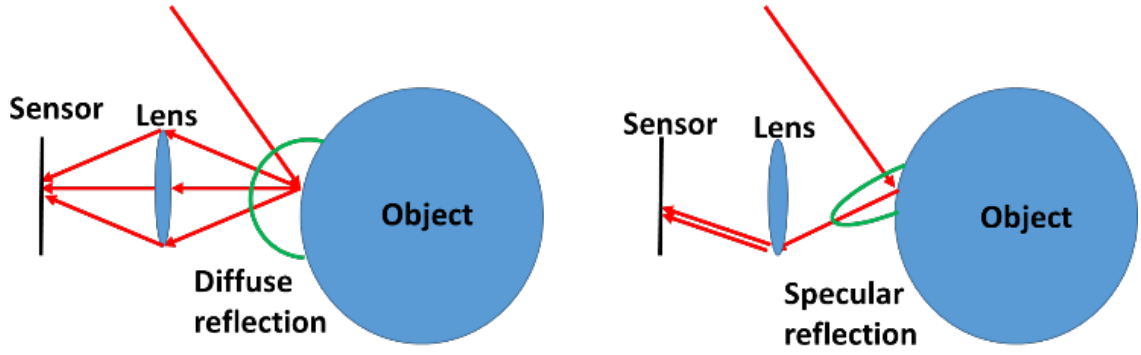


Figure 4.5: Diffuse versus specular reflection. Notice how the image sensor only receives rays from one specific angle in specular reflection which violates the Lambertian assumptions often made in computer vision algorithms.

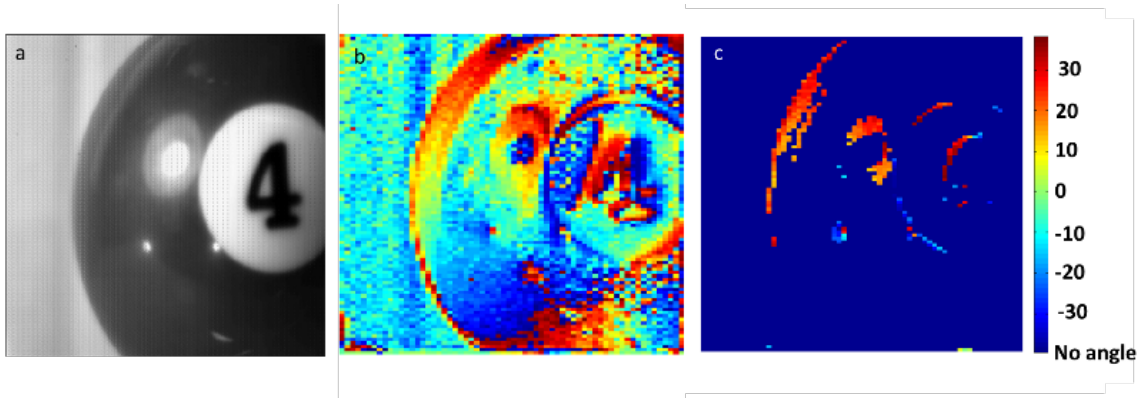


Figure 4.6: a) Scene with specular reflection, b) shows the angle of polarization when no thresholding is performed, and c) shows specular highlights tagged with a threshold of 40mV.

ular reflection within scenes. Light undergoing refraction in dielectric materials reflect a polarized component that appears as a strong highlight. This specular reflection can lead to difficulty for computer vision and light field algorithms that make the modeling assumption of Lambertian surfaces with diffuse reflectance as shown in Figure 4.5. Automatic tagging and reducing specular reflection using polarization can allow for better scene understanding and reconstruction in vision algorithms [139].

In Figure 4.6, we show an image of a scene containing a pool ball with specular high-

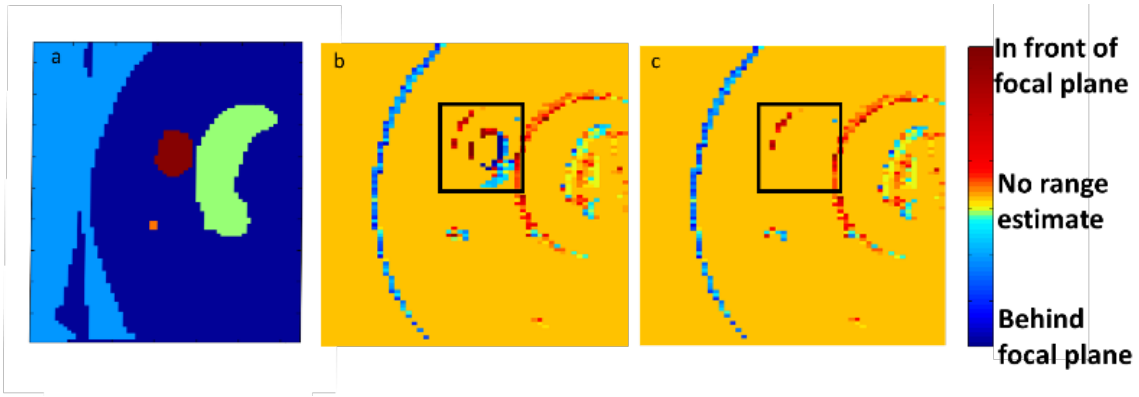


Figure 4.7: a) shows image segmentation performed on the original imaging, b) shows the depth map computed from light field captured by ASPs, and c) shows the removal of inaccurate depths at the specular locations

lights and the image with specular reflections identified and tagged using the measured angle of polarization. We use differences in polarization images I0-I90 and I45-I135 to detect specular reflection, and we use a threshold operation (around 30-40 mV) to avoid noise and low signal. One can note that the algorithms do not misidentify saturated pixels due to bright light as specular reflection, so the algorithm is robust even in those conditions. One main problem with using a tile of polarization sensors is that intensity variations across the tile due to edges can be mislabeled as changes in polarization. We reduce this effect in our image processing by selecting pixels close to one another for polarization calculations and correcting for the image gradient in a tile due to edges. We note that more advanced methods can be used as in [58].

In Figure 4.7, we show a depth map computed from light field information captured by ASPs similar to [196]. Note that light field algorithms generally only estimate depth from edges in the scene. It is clear that the depth estimates fail on specular reflection. In Figure 4.7, we show how using image segmentation and specular reflection tagging can remove these problem regions and lead to better depth accuracy of ASPs. We use image

segmentation in the original image via thresholding and active contours, and then remove segments with a majority of pixels that are specular to remove specular highlights from the scene. This shows the advantage of dual light field and polarization imaging for specular reflection: the polarization information can enhance the performance and robustness of light field algorithms.

4.3 Conclusions and Future Work

We have shown and characterized the polarization response of ASPs. There are interesting tradeoffs between the design of gratings for good diffraction response (corresponding to the modulation efficiency) and the extinction ratio for polarization. Polarization can be modeled in light field imaging as an additional sinusoidal modulation multiplied to the pixel response. We showed the ability to image stress-induced birefringence using ASPs, and showed how automatic specular reflection tagging can improve traditional light field depth algorithms.

Future applications include utilizing the lensless capabilities of ASPs with polarization for enhanced microscopy. In addition, there has been renewed interest in shape-from-polarization algorithms where polarization cues have improved depth maps from a Kinect TOF sensor [94]. One could imaging using polarization cues to improve depth maps from light fields without using an active light source. However, the current low polarization sensitivity of ASP pixels prevents the practicality of this application, especially for diffuse polarization.

CHAPTER 5

TIME-OF-FLIGHT

As stated earlier, time-of-flight (TOF) is not strictly a dimension of the plenoptic function, but couples depth of an object z with time t. However as an imaging technology, time-of-flight has quickly emerged into modern applications including autonomous vehicles, robotics, and augmented/virtual reality. In this chapter[1] we explore the possibility of combining TOF imaging with plenoptic imaging (in particular light field imaging). particularly the feasibility of an ASP on-chip implementation as a single hybrid depth sensor.

To introduce combined TOF and light field imaging, we introduce the conceptual framework of a depth field, a 4D spatio-angular function where the output is not radiance, but depth (via optical path length). Depth fields combine light field advantages such as synthetic aperture refocusing with TOF imaging advantages such as high depth resolution and coded signal processing to resolve multipath interference. We show applications including synthesizing virtual apertures for TOF imaging, improved depth mapping through partial and scattering occluders, and single frequency TOF phase unwrapping. Utilizing space, angle, and temporal coding, depth fields can improve depth sensing in the wild and generate new insights into the dimensions of light's plenoptic function.

5.1 Motivation for Depth Field Imaging

The introduction of depth sensing to capture 3D information has led to its ubiquitous use in imaging and camera systems, and has been a major focus of research in computer vision and graphics. Depth values enable easier scene understanding and modeling which in turn

[1] Most of the work in this chapter was originally presented in S. Jayasuriya et al., "Depth fields: extending light field techniques to time-of-flight imaging", 3DV 2015 [90].

Feature	Stereo	Photometric Stereo	Structured Illumination	Light Field	Time-of-Flight	Depth Fields (proposed)
On-chip pixel implementation	No	No	No	Yes	Yes	Yes
Illumination source	Passive	Active	Active	Passive	Active	Active
High resolution depth maps	No	Yes	Yes	No	Yes	Yes
Texture needed for depth	Yes	No	No	Yes	No	No
Ambiguity in depth measurement	No	Yes	No	No	Yes	No

Table 5.1: Table that summarizes the relative advantages and disadvantages of different depth sensing modalities including the proposed depth fields.

can realize new computer vision systems and human-computer interaction. Many methods have been proposed to capture depth information such as stereo, photometric stereo, structured illumination, light field, RGB-D, and TOF imaging.

However depth cameras typically support only one depth sensing technology at a time which limits their robustness and flexibility. Each imaging modality has its own advantages and disadvantages for attributes such as on-chip implementation, cost, depth resolution, etc that are summarized in Table 5.1. We argue that hybrid 3D imaging systems which utilize two or more depth sensing techniques can overcome these individual limitations. Furthermore, a system that combines modalities with an on-chip implementation would be cost effective and mass producible, allowing ubiquitous robust depth sensing.

We propose combining light field and TOF imaging into a hybrid 3D imaging system. This system inherits light field advantages such as post-capture digital refocusing with TOF advantages of high resolution depth information and the mitigated multipath interference

using coded signals. Further, light field and TOF imaging both have been implemented on-chip [62, 141], and we can design hybrid pixel structures to combine both modalities on-chip as well. Each modality has its relative disadvantages: depth from light fields require textured surfaces and is dependent on object distance for disparity, and single frequency TOF imaging suffers from phase wrapping and is limited to small aperture cameras with low shutter speeds. However, we show that combining light field and TOF imaging can alleviate all of these limitations.

We call this extension of spatio-angular information captured traditionally by light fields to TOF depth maps as **depth field imaging**. Our main contributions include:

- Formulation of depth field imaging as an extension of the light field framework for TOF imaging
- Methods to capture depth fields using camera arrays and single-shot camera systems.

We show that capturing depth fields leads to many new applications that improve robust depth sensing in the wild including:

- Digital refocusing of depth images and extended depth of field.
- Phase unwrapping for single frequency TOF imaging.
- Depth imaging through partial occluders.
- Depth imaging and refocusing past scattering media.

A larger vision for introducing depth fields is to have a layer of post-capture control for depth sensing which can combine synergistically with higher level algorithms such as structure from motion (SfM) [40], Simultaneous Localization and Mapping (SLAM) [180],

and reconstructions from collections of online images [170] used for 3D reconstruction and scene modeling/understanding.

5.2 Related Work

We survey related work in LF imaging, TOF imaging, and fusion algorithms for depth imaging to show the context of depth sensing technologies that depth field imaging relates to.

Light Field Imaging captures 4D representations of the plenoptic function parametrized by two spatial coordinates and two angular coordinates, or equivalently as the space of non-occluded rays in a scene [65, 121]. Light fields are used for image-based rendering and modeling, synthesizing new viewpoints from a scene, and estimating depth from epipolar geometry. In the context of cameras, light fields have been captured by using mechanical gantries [120] or large dense camera arrays [204], or by single-shot methods including microlenses [2, 141], coded apertures [119], transmission masks [186], or diffraction gratings [84]. Light fields can extend the depth of field and use digital refocusing to synthesize different apertures in post-processing [141], thus enabling a level of software control after the photograph has been taken. We will exploit this control in depth field imaging.

Time-of-Flight Imaging works by encoding optical path length traveled by amplitude modulated light which is recovered by various devices including photogates and photonic mixer devices [9, 62, 110, 162]. While yielding high resolution depth maps, single frequency TOF suffers from limitations including phase wrapping ambiguity and multipath interference caused by translucent objects and scattering media. Proposed techniques to

overcome these limitations include phase unwrapping with multifrequency methods [149], global/direct illumination separation [93, 207], deblurring and superresolution [209], and mitigating multipath interference with post-processing algorithms [15, 16]. Recently, new temporal coding patterns for these sensors help resolve multiple optical paths to enable seeing light in flight and looking through turbid media [79, 80, 95]. Similar to our work, camera systems have been proposed to fuse together TOF + stereo [219], TOF + photometric stereo [181], and TOF + polarization [94].

Fusion of depth maps and intensity images has been used to enable 3D reconstruction by explicit feature detection [81, 85]. Real-time interaction for camera tracking and 3D reconstruction have been demonstrated via KinectFusion [88]. While conceptually similar to depth fields by acquiring per-pixel values of depth and intensity, these fusion methods do not systematically control the spatio-angular sampling or transcend the traditional capture tradeoffs between aperture and depth of field for depth imaging. In this way, we hope that depth field algorithms can serve as the foundation upon which fusion algorithms can improve their reconstruction quality, leading vertical integration from camera control all the way to high level scene modeling and understanding.

5.3 Depth Fields

In this section, we combine the mathematical formulations of light field and TOF imaging into the concept of a depth field. We show both how to capture these fields and how to invert the forward model to recover light albedo, defined as the reflectance value of an object with respect to the active illumination, and depth as a function of 2D spatial coordinates and 2D angular coordinates. This approach is similar to Kim et al. [99] who capture depth

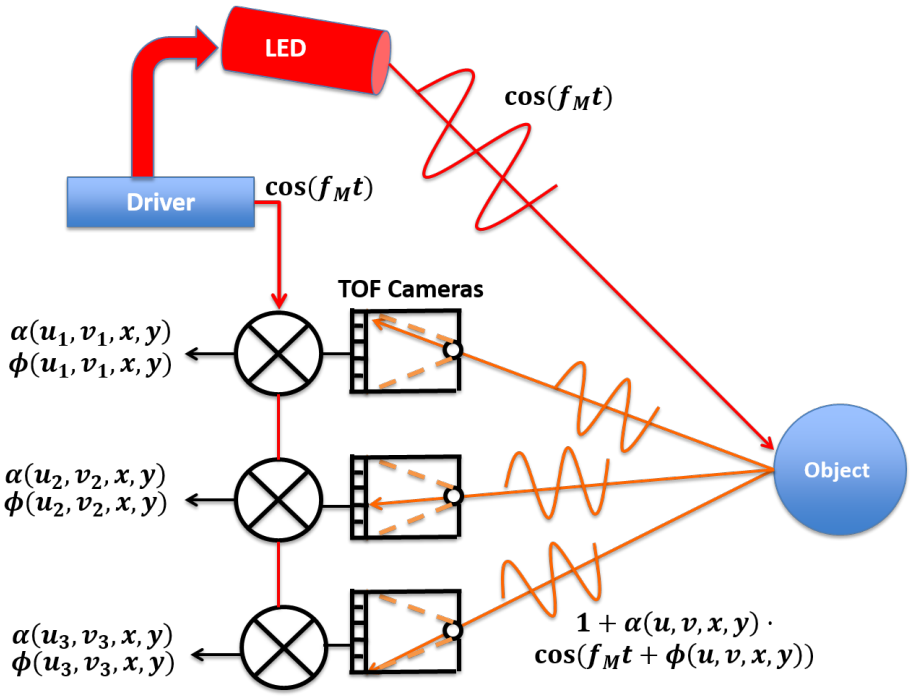


Figure 5.1: Capturing a depth field conceptually using an array of TOF cameras

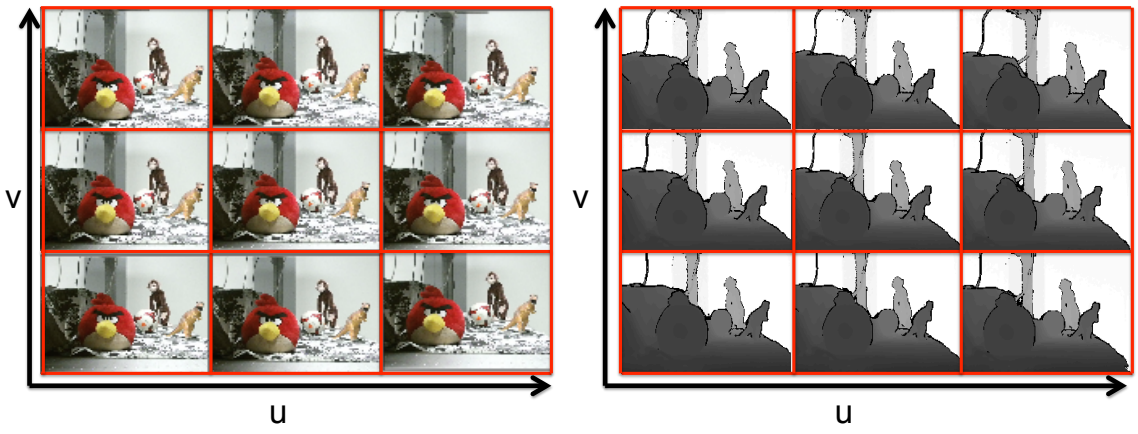


Figure 5.2: Depth field as a 4D function of albedo and phase.

maps for different perspective views, but they do not use TOF imaging or show applications such as digital refocusing or depth mapping through partial/scattering occluders.

To describe the forward model for capturing depth fields, we first briefly discuss the forward models for light field and TOF imaging.

5.3.1 Light Fields

Light fields are commonly parameterized by the two plane model $l(u, v, x, y)$ where (u, v) is the angular coordinates at the lens plane, and (x, y) are the spatial coordinates of the sensor plane [121]. The output of this function represents the radiance of the ray parametrized by its intersection with the two planes. The forward model for light field capture has been modeled in [202] as follows:

$$i_{LF}(x, y) = \int_u \int_v m(u, v, x, y) \cdot l(u, v, x, y) du dv \tag{5.1}$$

where $i_{LF}(x, y)$ is the intensity measured by the detector and $m(u, v, x, y)$ is the modulation/multiplexing function that encodes the incoming light rays. The modulation function represents the different optical elements that could be used to sense the light field including pinholes ($m(u, v, x, y) = \delta(u, v, x, y)$), Fourier masks, random codes/masks, or diffraction gratings where the modulation functions are Gabor wavelets [84]. Discretizing the above equation, $\mathbf{i}_{LF} = \mathbf{M}\mathbf{l}$ where $\mathbf{i}_{LF}, \mathbf{l}$ are the vectorized images and light fields, and $\mathbf{M}$ is the modulation matrix, and both linear and nonlinear inversions can recover back the light field [202].

5.3.2 Time-of-Flight Imaging

In contrast, TOF is typically modeled using a cross-correlation between the incoming light signal and the reference code sent to the sensor. Given that incoming light is of the form: $1 + \alpha \cos(f_M t + \phi(x, y))$ where ϕ is the phase accumulated due to the optical path traveled from light source to object to camera and α is the albedo, the intensity at the sensor (normalized to integration time) is:

$$\begin{aligned} i_{TOF}(\tau, x, y) &= (1 + \alpha(x, y) \cos(f_M t + \phi(x, y))) \otimes cos(f_M t) \\ &\approx \frac{\alpha(x, y)}{2} \cos(f_M \tau + \phi(x, y)). \end{aligned} \tag{5.2}$$

Here, τ is the cross-correlation parameter which controls the phase shift of the reference signal. By choosing different τ such that $f_M \tau = 0, \pi/2, \pi, 3\pi/2$, we can recover both the albedo α and the phase ϕ at each spatial location (x,y) using quadrature inversion:

$$\begin{aligned} \phi(x, y) &= \tan^{-1}((i_{TOF}(\frac{3\pi}{2}) - i_{TOF}(\frac{\pi}{2}))/(i_{TOF}(\pi) - i_{TOF}(0))), \\ \alpha &= \sqrt{(i_{TOF}(\frac{3\pi}{2}) - i_{TOF}(\frac{\pi}{2}))^2 + (i_{TOF}(\pi) - i_{TOF}(0))^2}. \end{aligned} \tag{5.3}$$

Note that $d = \frac{c \cdot \phi}{4\pi f_M}$ can directly recover depth d from phase ϕ for TOF imaging.

5.3.3 Depth Fields

We now introduce the concept of the **depth field** as the ordered pair of albedo and depth (encoded in phase) (α, ϕ) that occurs at every (u, v, x, y) spatio-angular coordinate, i.e. $\alpha = \alpha(u, v, x, y), \phi = \phi(u, v, x, y)$. Note that depth fields are not recoverable from TOF measurements alone since TOF assumes a pinhole camera model, which sample ϕ and

α at a particular fixed (u, v). We now describe the forward model of depth field imaging as follows:

$$i(\tau, x, y) = \int_{u,v} m(u, v, x, y) \cdot (1 + \alpha(u, v, x, y) \cos(f_M t + \phi(u, v, x, y))) du dv \otimes \cos(f_M t) \tag{5.4}$$

which is approximately

$$i(\tau, x, y) \approx \int_{u,v} m(u, v, x, y) \cdot \frac{\alpha(u, v, x, y)}{2} \cdot \cos(f_M \tau + \phi(u, v, x, y)) du dv. \tag{5.5}$$

To invert this model, we take four measurements $f_M \tau = 0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}$ to get images $i(0), i(90), i(180), i(270)$ at each spatial location. Then we calculate $\mathbf{M}^{-1} i(\tau)$ to invert the light field matrix for each of these images (Note: this inverse can be either done at lower spatial resolution or using sparse priors or modeling assumptions to retain resolution). Thus we recover albedo and phase mixed together at every (u, v, x, y):

$$D' = \frac{\alpha(u, v, x, y)}{2} \cdot \cos(f_M \tau + \phi(u, v, x, y)). \tag{5.6}$$

To unmix the albedo and phase, we can perform quadrature inversion on D' for $f_M \tau = 0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}$ as before in TOF to recover the depth field.

5.4 Methods to Capture Depth Fields

5.4.1 Pixel Designs

We describe the potential for single-shot capture of depth fields (Note: single-shot is a misnomer since 4 phase measurements are performed per shot, however such function-

ality can be built into hardware to work in a single exposure). In particular, we outline the design of these concept pixels in Figure 5.3. The only image sensor fabricated to date capable of capturing depth fields in a single shot (that we know of) integrates metal diffraction gratings over single photon avalanche diodes [116], however this sensor is used for lensless fluorescence imaging. All these single-shot methods sacrifice spatial resolution to multiplex the incoming depth field.

As in most light field sensors, we can align microlenses above CMOS TOF sensors such as photogates, photonic mixer devices, etc. Doing so allows sampling the angular plane by sacrificing spatial resolution at the sensor plane. The main lens can widen its aperture, allowing more light transmission while each of the sub-aperture views underneath the microlenses maintains a large depth of field [141]. This is advantageous since existing TOF cameras sacrifice exposure time to keep a small aperture and large depth of field. One limitation is the need for fine optical alignment of the microlenses at the conjugate image plane in the camera body.

Another depth field sensor can use amplitude masks between the main lens and the sensor plane of photogates to filter incoming angular rays [186]. While allowing less light transmission as microlenses, masks can be designed with different coding patterns for improved reconstruction of the depth field and can be flexibly interchanged within the camera body unlike fixed optical elements. We note a similar technique from [61] which uses a coded aperture in front of LIDAR system to extend the system's depth of field.

5.4.2 Angle Sensitive Photogates

We also propose a fully integrated CMOS pixel design that does not require alignment of external optical elements: integrated diffraction gratings over interleaved photogates similar to [169]. We call these sensors Angle Sensitive Photogates to differentiate from regular ASP pixels. Note that this pixel can achieve better light efficiency with phase gratings and reduce its pixel size with interleaved photogates while maintaining the advantages of CMOS integration for cost and mass-production.

We further designed these specialized pixels in a 130nm BiCMOS process, and show the final layout in FIgure 5.4. We couldn't fabricate the interleaved photogate structure due to design rule checks from the foundtry, but this is a practical limitation that can be overcome by a custom process for these pixels. While the design of these pixels was straightfoward, designing the support circuitry and timing for the amplifiers/readout circuitry is fairly complex. We leveraged a modern digital hardware flow in order to do so (see Appendix A for further details).

5.4.3 Experimental Setup

Since fabricating a CMOS sensor takes significant time and resources, we motivate the need for depth field imaging using a custom acquisition setup. This acquisition setup captures depth fields at high spatial resolution by moving a TOF camera on a two axis stage sequentially in the (u, v) plane, as if having an array of TOF cameras, to scan a depth field. See Figure 5.2 for a schematic depiction of depth field capture.

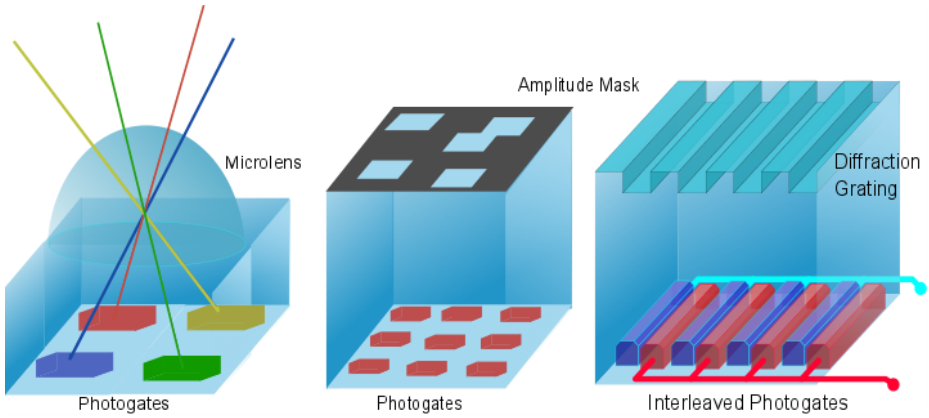


Figure 5.3: Pixel designs for single-shot camera systems for capturing depth fields. Microlenses, amplitude masks, or diffraction gratings are placed over top of photogates to capture light field and TOF information simultaneously.

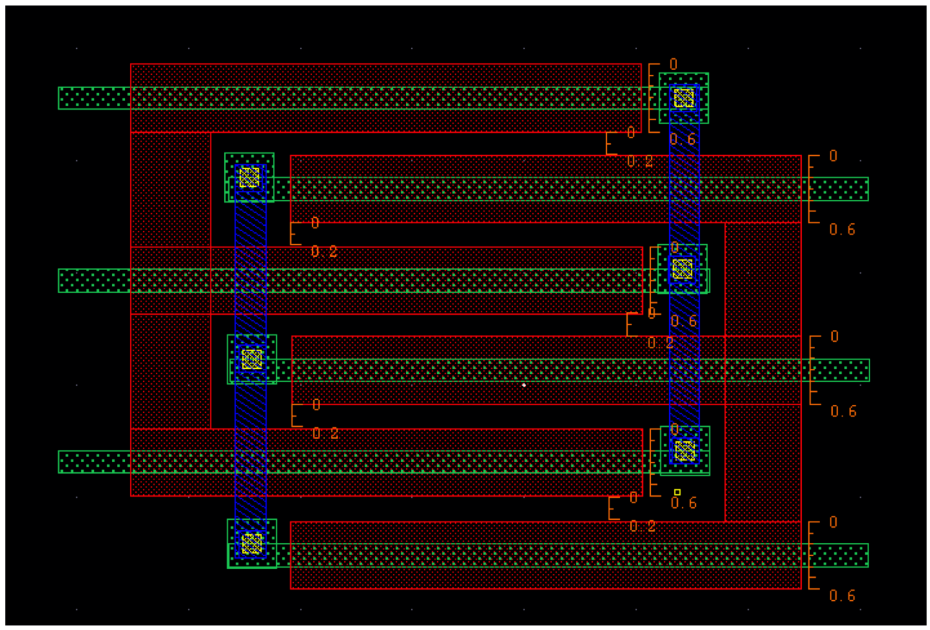


Figure 5.4: Pixel layout for an Angle Sensitive Photogate. Green = diffusion, Red = polysilicon, Blue = M1.

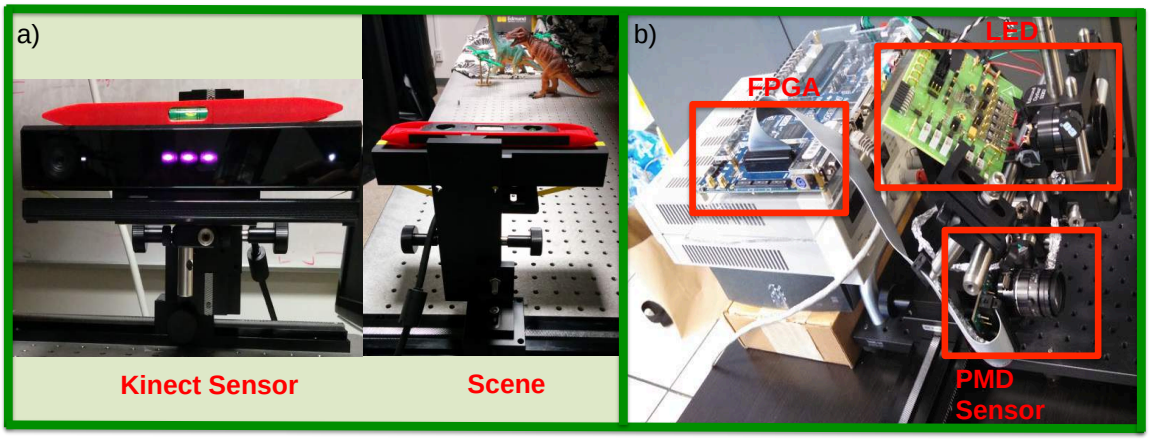


Figure 5.5: Setup to capture depth fields in practice. (a) A Kinect is placed on a XY translation stage on an optical bench, and a representative imaging scene, (b) PMD sensor with FPGA for code generation and LED setup as in [95]

5.5 Experimental Setup

In this section, we describe the depth field acquistion setup that scans a TOF sensor. We choose this approach since existing TOF sensors have limited resolution so single-shot methods would result in even smaller spatial resolution and the need of precise alignment of microlenses or masks. Our system has some limitations including a bulky setup and static scene acquisition, but we still demonstrate advantages of depth field imaging.

We move a TOF sensor on a two axis stage at different (u, v) positions. We utilize both the Microsoft Kinect One which has a 424 x 512 depth resolution (see Figure 5.5), and a custom PMD sensor of 160 x 120 resolution which enables us to send custom modulation codes directly to the silicon. The PMD sensor setup is the same as that described in [95]. All depth fields were captured with 1" spacing in the (u, v) plane at the coarse resolution of 5x5.

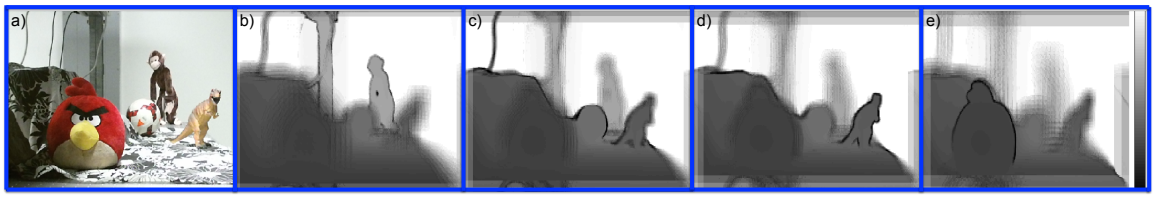


Figure 5.6: a) Captured scene, b-e) Digital refocusing on different focal planes for the depth map of the scene, showing how depth field imaging can break the tradeoff between aperture and depth of field for range imaging

5.6 Applications of Depth Fields

In this section, we highlight new applications of depth field imaging.

5.6.1 Synthetic Aperture Refocusing

One main disadvantage of TOF imaging is the necessity of a small aperture for large depth of field to yield accurate depth values. Having a shallow depth of field or wide aperture causes optical blur which corrupts TOF depth values. However, a small aperture limits the shutter speed and increases the acquisition time for these systems. In contrast, light field imaging breaks this tradeoff between depth of field and aperture size by using synthetic aperture refocusing. A plenoptic sensor with microlenses above its pixels can open its aperture and allow more light transmission while keeping the sub-aperture images beneath the microlenses in-focus, albeit at the loss of spatial resolution. After capture, one can digitally refocus the image, thus extending the depth of field by shearing the 4D light field and then summing over (u, v) to synthesize images with different focal planes [141].

Similarly, we show that the same techniques can be applied to depth fields. In Figure 5.6, we show digital refocusing of the 4D $\phi(u, v, x, y)$ information by applying the same

shear and then average operation [141]. We are able to synthesize capture through a large virtual aperture for the scene which has not been shown in depth maps before, and may be combined with wide aperture light intensity images for enhanced artistic/photographic effect. In addition, this validates that single-shot depth field sensors such as TOF sensor with microlenses can allow more light through the aperture, thus increasing exposure while maintaining the same depth of field. This enables decreased acquisition time for TOF sensors at the expense of computationally recovering the lost spatial resolution and depth of field in post-processing algorithms. We note that [61] also showed extended depth of field for a LIDAR system using a coded aperture, but they don't extend their framework to show applications such as digital refocusing.

5.6.2 Phase wrapping ambiguities

One main limitation for single frequency TOF is that the phase has 2π periodicity, and thus depth estimates will wrap around the modulation wavelength. For modulation frequencies in the tens of MHz, this corresponds to a depth range of a few meters, which can be extended further by using multiple frequencies [44, 149] or phase unwrapping algorithms [45]. However, as modulation frequencies scale higher, phase wrapping becomes more severe.

We observe that capturing depth fields at a single modulation frequency also allows us to unwrap the phase periodicity by utilizing inherent epipolar geometry from different viewpoints. We use the depth from correspondence algorithm from [178] which is coarse and distance dependent, but does not suffer from phase wrapping, and thus can unwrap the depth measurements given by TOF.

In Figure 5.7, we simulate the Cornell Box scene and capture a depth field using the ray tracer Mitsuba [89]. We simulate phase wrapping and calculate depth from correspondence. In order to perform phase unwrapping, we select a continuous line in the image (the side wall in this scene) to determine the number of times the TOF image wraps upon itself in the scene. We use this mapping to match the wrapped TOF depth values to the depth values from correspondence, leading to unwrapped TOF depth values for the entire image as shown in Figure 5.7d. We also use a median filter to alleviate edge discontinuities in calculating depth from correspondence.

In the Microsoft Kinect we use for capturing depth fields, the maximum modulation frequency is 30MHz, which makes showing phase wrapping difficult on a standard optical bench. Thus we change the bit settings on the Kinect TOF sensors from N bits to N-1 bits to simulate phase wrapping for a real scene (identical to the wrapping caused by periodicity at a higher modulation frequency of 60MHz). We show the results of our phase unwrapping algorithm in Figure 5.8. Note that the reconstruction quality is limited by the lack of a good fiducial line in the scene that clearly corresponds light field depths to TOF wrapped depths. This is a limitation of our method, and it would be interesting to explore automatic calibration for phase unwrapping.

5.6.3 Refocusing through partial occluders

The large synthetic aperture that can be synthesized by capturing 4D depth fields allows us to image past partial occluders in the foreground. This technique, which blurs out the foreground to reveal the background, has been shown in light fields [184] to look through bushes and plants. Note that in Figure 5.9, applying the same technique to the depth field works correctly for the albedo (one can see the object clearly while blurring out

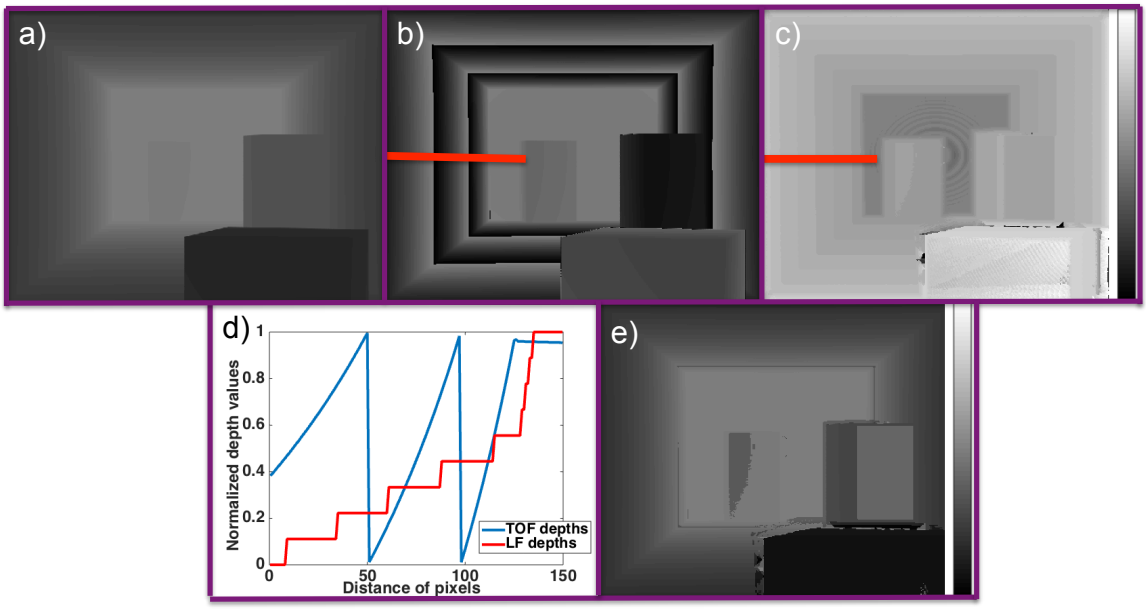


Figure 5.7: Phase unwrapping algorithm on synthetic data. a) Cornell box scene with ground truth depth values, b) a phase wrapped scene with red fiducial line for calibration marked, c) depth map given by light field correspondence algorithm. We identify the same calibration line in this scene for phase unwrapping, d) we map the TOF wrapped values to the depth values from correspondence for the given calibration line, e) unwrapped depth map.

the foreground), but it does not work for the phase. This is because while visually we can perceptually tolerate some mixing of foreground and background color, this same mixing corrupts our phase measurements, leading to inaccurate depth values.

To solve this mixing problem when refocusing light fields, researchers have simply not added rays that are from the foreground when averaging over the sheared light field. A key assumption to their algorithm is that the foreground object rays are identified either by shooting continuous video [204] or by constructing an epipolar image, finding the corresponding depths, and then separating foreground relative to the background [184]. These algorithms are computationally expensive to identify the occluding objects pixels. We note that [212] use a combination of unstructured multiview stereo views and a depth sensor to refocus an intensity image through a partial occluder, and use the depth information to

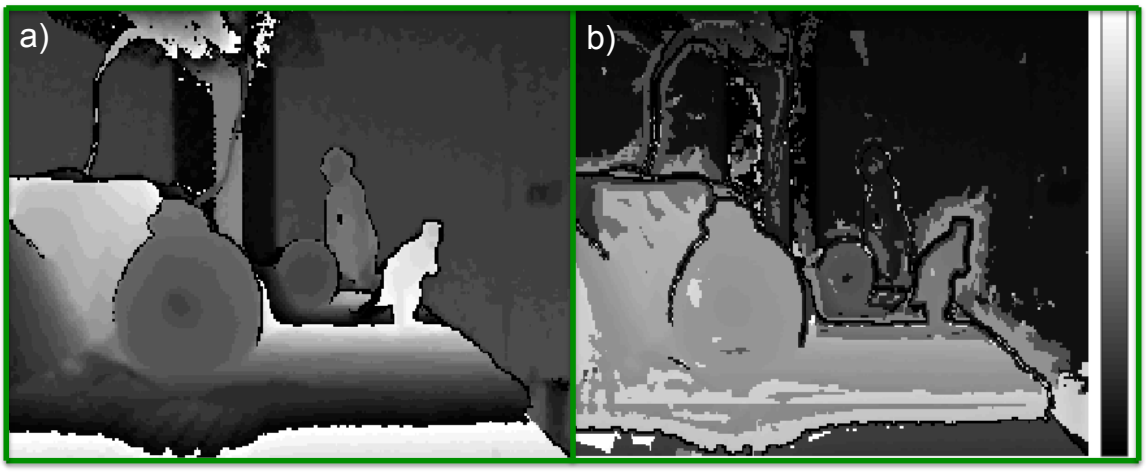


Figure 5.8: a) Phase unwrapping on real data with synthetic phase wrapping induced (due to prototype limitations). b) Recovered depth map. Notice that the monkey in back is not recovered because there does not exist a calibration marker line in the scene that extends all the way back in the TOF image.

create a probabilistic model for occluders.

In contrast, we utilize the depths directly captured via TOF measurements to construct a histogram of depths observed in the scene as shown in Figure 5.9. We then can simply pick a foreground cluster using K-means or another computationally efficient clustering algorithm, which is faster than constructing an epipolar image, estimating line slopes, and then forming a histogram to do clustering. In Figure 5.9, you can see the results of our algorithm.

5.6.4 Refocusing past scattering media

While the previous subsection dealt with partial occluders that block the background for certain (u, v) viewpoints, other occluders such as scattering media or translucent objects are more difficult because they mix multiple phase measurements corresponding to

different optical path lengths together at a single pixel. We approach the problem via coded TOF, specifically the depth selective codes by [175]. Mainly, we show how coded TOF extends the capabilities of our depth field camera systems by imaging past scattering media, and then use spatial information to perform digital refocusing. In Figure 5.10, we image through backscattering nets to get a depth field past the scattering media. We place nets in front of the camera to act as strong backscatterers, notice how the depth values are corrupted by the scattering. Using the depth selective codes, we can image past the nets, and using multiple shots at different (u, v) viewpoints, we can capture the depth field beyond the nets and do digital refocusing. This demonstrates how depth field imaging can leverage the advantages of coded TOF techniques, and poses interesting questions of how to design the best possible codes for single-shot depth field imaging systems.

5.7 Discussion

Depth fields unify light field and TOF imaging as a single function of spatio-angular coordinates, and are useful for various applications. Besides the simple extensions of adding two imaging modalities, they can inform each other and make algorithms computationally more efficient and conceptually simpler, particularly in solving the problem of various occluders for light fields by using TOF information and breaking tradeoffs between aperture and depth of field for TOF cameras by adding light field capability. Improvements in light field depth estimation such as in [178] can also be applied for depth field cameras leading to improved depth resolution.

A key question that concerns depth field cameras is their pixel size which makes pixel multiplexing, including the sensor designs outlined in this chapter, problematic. We note

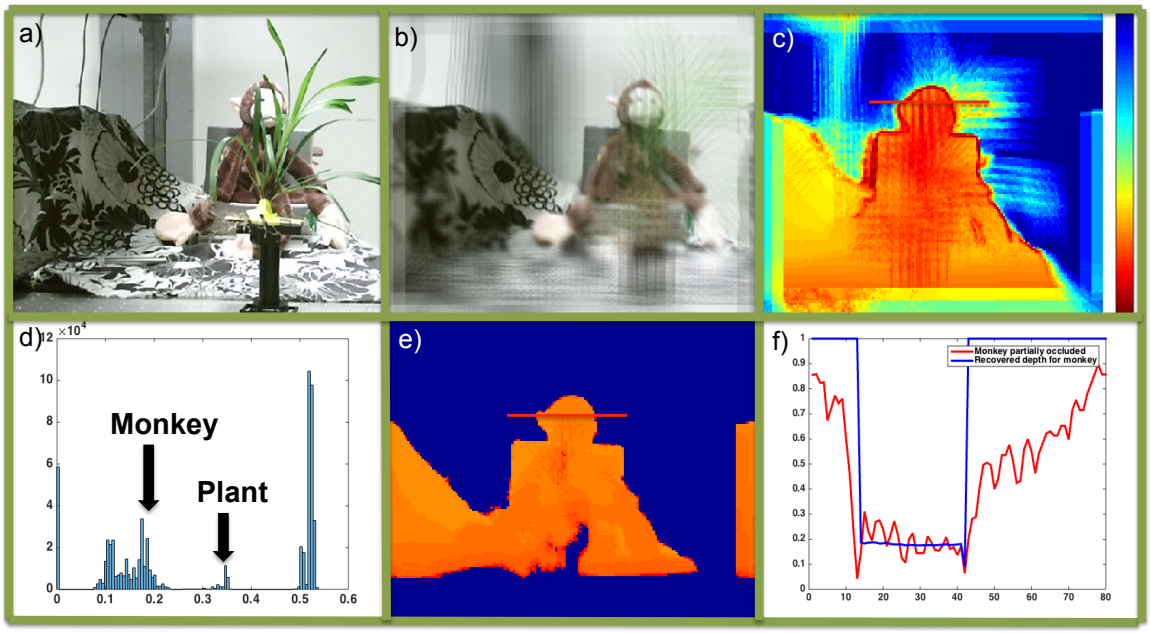


Figure 5.9: Refocusing in spite of foreground occlusions: (a) Scene containing a monkey toy being partially occluded by a plant in the foreground, (b) traditional synthetic aperture refocusing on light field is partially effective in removing the effect of foreground plants, (c) synthetic aperture refocusing of depth displays corruption due to occlusion, (d) histogram of depth clearly shows two clusters corresponding to plant and monkey, (e) virtual aperture refocusing after removal of plant pixels shows sharp depth image of monkey, (f) Quantitative comparison of indicated scan line of the monkey's head for (c) and (e)

that TOF pixels have shrunk currently to 10um [9] which is only 10x larger than regular pixels (1um), and that technological advances such as stacked image sensors may help alleviate these multiplexing worries. However, the clear advantages for depth field cameras are applications where spatial resolution is not the limiting factor. This includes imaging systems that are limited by aperture (as argued in Section 6.1) and lensless imaging where spatial pixel layout is not a factor [59].

5.7.1 Limitations

Some limitations include long computational algorithms to recover lost spatial resolution for single-shot depth field cameras, or increased acquisition time for large TOF camera arrays or TOF cameras on mechanical gantries to scanline a depth field. Many applications provide partial robustness to depth sensing in the wild, but rely on modeling assumptions (foreground vs. background separation, scattering media is not immersing the object) that limits their deployment in real autonomous systems.

5.8 Conclusions and Future Work

In this chapter, we have laid the preliminary work for incorporating TOF imaging into the plenoptic imaging framework. To do so, we introduce the depth field, a representation of light albedo and phase from optical path length as function of space and angle. We developed the forward model to inform new ways to capture depth fields, and showed a myriad of applications that having this information allows.

One main future direction is to fabricate CMOS photogates with integrated diffraction

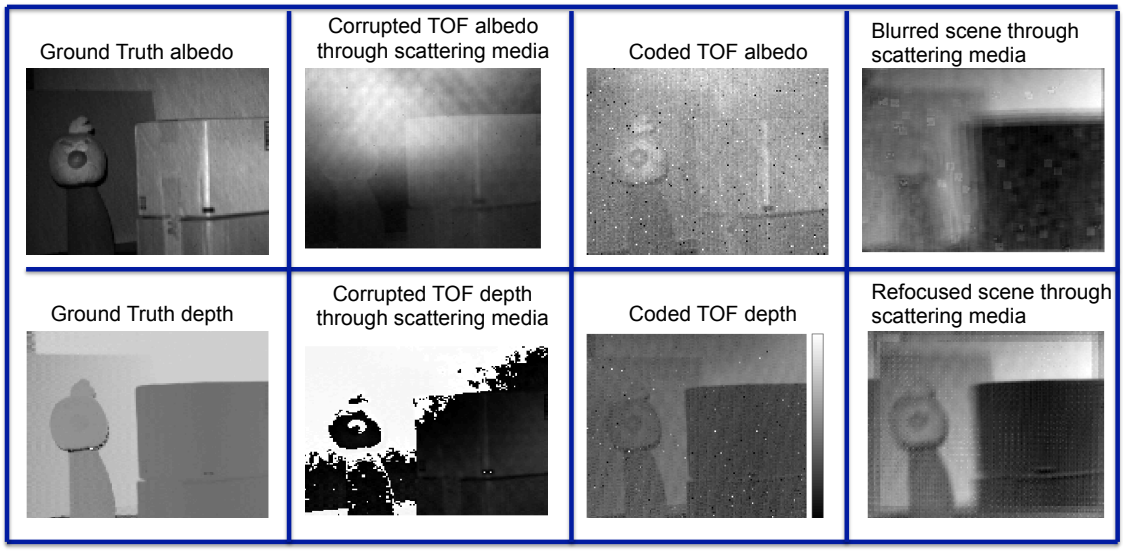


Figure 5.10: We use coding techniques from [175] to image beyond backscattering nets. Notice how the corrupted depth maps are improved using the codes. We show how digital refocusing can be performed on the images without the scattering occluders by combining depth fields with coded TOF.

gratings for an on-chip depth field sensor. This chip would be a lensless depth sensor for biomedical or 3D printing applications where a physical lens is bulky and prevents deployment of the image sensor in confined locations. Our analysis of a depth field shows that one can invert a farfield image using angular resolution similar to lensless lightfield cameras [59].

However, designing computational image sensors is a difficult task, requiring expertise in analog and mixed-signal design for the sensors, and digital VLSI and computer architecture to build real systems that can perform computation on-board the sensor. This work directly led to the development of a new hardware design flow that aims to make vertically-integrated hardware research possible. We encourage the reader to see Appendix A for a detailed description of this design flow.

CHAPTER 6

VISUAL RECOGNITION

While the previous chapters have primarily incorporated new plenoptic dimensions into ASP imaging, the goal of this chapter[1] is to leverage the plenoptic information provided by ASPs to perform visual recognition tasks. In particular, we show that ASPs can optically compute the first layer of convolutional neural networks, state-of-the-art deep learning algorithms which are surpassing humans in some object identification tasks. We explore the energy savings by doing so, and showcase real digit and face recognition from our prototype ASP setup.

6.1 Introduction

State-of-the-art visual recognition algorithms utilize convolutional neural networks (CNNs) which use hierarchical layers of feature computation to discriminate visual stimuli. Early CNNs from LeCun et al. [113] showed promising results in digit recognition [114]. The advent of GPU computing has allowed CNN training on large, public online data sets, and triggered an explosion of current research. CNNs have started to perform on par with or even surpass humans on some image recognition challenges such as ImageNet [106, 156]. CNNs have been universally applied to different vision tasks including object detection and localization [174], pedestrian detection [173], face recognition [161], and even synthesizing new objects [43]. However, many applications in embedded vision such as vision for mobile platforms, autonomous vehicles/robots, and wireless sensor networks have stringent constraints on power and bandwidth, limiting the deployment of CNNs in these contexts.

[1] This work was originally presented in H. Chen et al., "ASP Vision: Optically Computing the First Layer of Convolutional Neural Networks using Angle Sensitive Pixels", CVPR 2016 [33].

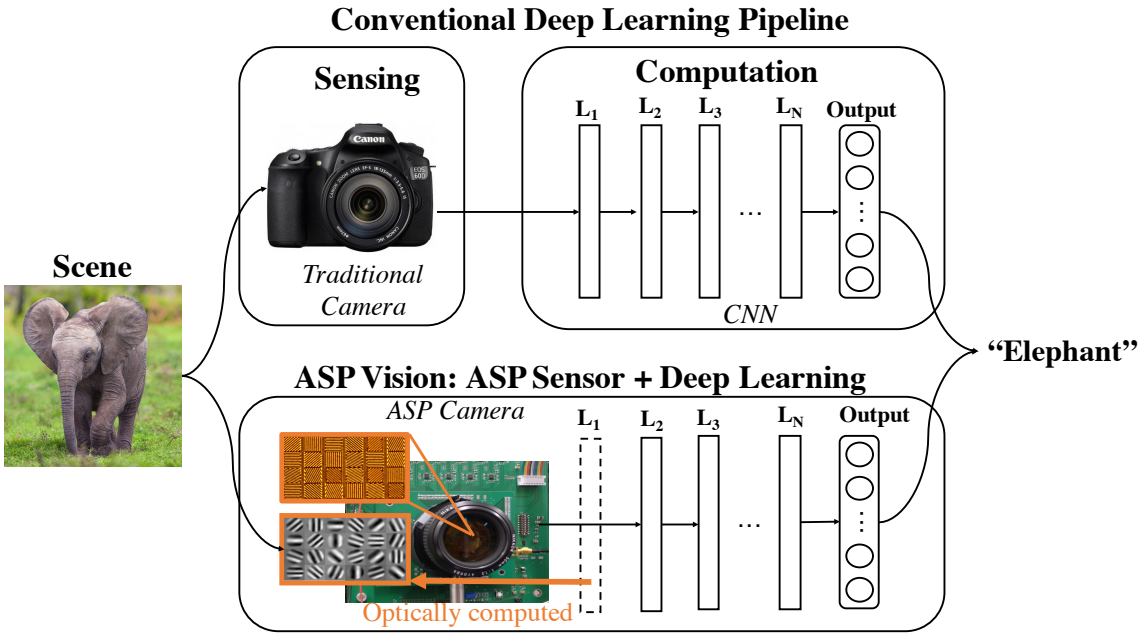


Figure 6.1: ASP Vision, our proposed system, is compared with a conventional deep learning pipeline. ASP Vision system saves energy and transmission bandwidth in the sensing stage, compared to a traditional camera.

6.1.1 Motivation and Challenges

Porting deep learning vision systems to embedded and battery-operated applications necessitates overcoming the following challenges:

- **Sensor power:** Image sensors are notoriously power-hungry, sometimes accounting for more than 50% of the power consumption in many embedded vision applications [125]. In addition, current image sensors are not optimized to significantly save power for such computer vision tasks [125]. Several researchers, most recently [124], have argued that always-on, battery-operated, embedded vision systems necessitate a complete redesign of the image sensor to maximize energy-efficiency.

- **Computing power:** CNNs, while providing enormous performance benefits, also suffer from significantly increased computational complexity. GPUs and multi-core processors are power hungry, and the number of FLOPS (floating point operations) for CNNs can easily be on the order of billions.
- **Data bandwidth:** Data bandwidth requirements place strict design constraints on traditional vision architectures. Moderate image resolution of 1 megapixel at 30 fps (frames per second) results in a bandwidth requirement of over 0.5 Gbps (Giga-bits per second). This can bottleneck I/O buses that transfer images off the sensor to the CPU and increases the power requirements, computational complexity, and memory for the system.

6.1.2 Our Proposed Solution

To solve the challenges described above, we explore novel image sensors that can save energy in an embedded vision pipeline. In particular, we use existing Angle Sensitive Pixels (ASPs) [196], bio-inspired CMOS image sensors that have Gabor wavelet impulse responses similar to those in the human visual cortex, to perform optical convolution for the CNN first layer. We call this combination of ASP sensor with CNN backend *ASP Vision*. This system addresses embedded deep learning challenges by:

- **Reducing sensor power** by replacing traditional image sensors with energy-efficient ASPs that only digitize edges in natural scenes.
- **Reducing computing power** by optically computing the first convolutional layer using ASPs, thus leaving subsequent network layers with reduced FLOPS to compute.

- **Reducing bandwidth** by relying on the inherent reduced bandwidth of ASP sensors encoding only edge responses.

6.1.3 Contributions

In this chapter, we will describe in detail our system for optically computing the first layer of CNNs. Note that we are neither introducing ASPs for the first time nor claiming a new CNN architecture. Instead, we are deploying ASPs to increase energy efficiency in an embedded vision pipeline while maintaining high accuracy. In particular, our main contributions in this paper include:

- Showing the optical response of Angle Sensitive Pixels emulates the first layer of CNNs
- Analysis of the energy and bandwidth efficiency of this optical computation
- Evaluation of system performance on multiple datasets: MNIST [114], CIFAR-10/100 [105], and PF-83 [13].
- An operational prototype of the ASP Vision system and real experimental results on digit recognition and face recognition using our prototype.

6.1.4 Limitations

Our proposed approach is also limited by some practical factors. While there are significant potential FLOPS savings from optical computation, our current prototype achieves

a modest fraction of these savings due to the prefabricated sensor's design choices. In addition, ASPs themselves are challenged with low light efficiency and reduced resolution that we address in detail in Section 6.4.2. Finally, our current hardware prototype has limited fidelity since it was not fabricated in an industrial CMOS image sensor process. We discuss this in Section 6.5. We caution readers from placing too much expectation on the visual quality of a research prototype camera, but hope the ideas presented inspire further research in novel cameras for computer vision systems.

6.2 Related Work

In this section, we survey the literature with particular focus on energy-efficient deep learning, computational cameras, and hardware-based embedded vision systems.

Convolutional Neural Networks are currently the subject of extensive research. A high level overview of CNNs is given by LeCun et al. [112]. Since we not improve CNN accuracy or propose new networks, we highlight recent work on real-time performance and resource efficiency. Ren et al. use faster R-CNN [155] to achieve millisecond execution time, enabling video frame rates for object detection. In addition, researchers have explored reducing floating point multiplications [128], quantization of weights in CNNs [63, 73], network compression [34], and trading off accuracy for FLOPs [161].

On the sensor side, **computational cameras** have emerged to expand the toolset of modern imaging systems. Cameras have been augmented to capture light fields [84, 141, 186], polarization [70], high dynamic range [163], and depth [9]. Similar to ASPs, cameras that compute features include on-chip image filtering [67, 142] or detect events [122].

Embedded vision has been spurred by advances in imaging technology and digital processing. For convolutional neural networks, analog ASICs [18], FPGAs [49], and neuromorphic chips [151] implement low power calculations with dedicated hardware accelerators. LiKamWa et al. [124] propose a new analog-to-digital converter for image sensors that performs CNN image classification directly to avoid the I/O bottleneck of sending high resolution images to the processor. Micro-vision sensors [104] perform optical edge filtering for computer vision on tight energy budgets. Similar to our work, inference/learning on coded sensor measurements from compressive sensing imaging has saved bandwidth/computation [86, 108, 131]. Dynamic Vision Sensors (DVS) have been used for face recognition while saving energy compared to conventional image sensors [136]. All this research forecasts higher levels of integration between deep learning and embedded vision in the future.

6.3 ASP Vision

A diagram of our proposed ASP Vision system is presented in Figure 6.1. The custom image sensor is composed of Angle Sensitive Pixels which optically computes the first layer of the CNN used for visual recognition tasks. In the following subsections, we describe how hardcoding the first layer is application-independent, ASP design, and how ASPs perform optical convolution with energy and bandwidth savings. Finally, we discuss current limitations with ASP design and imaging for embedded vision.

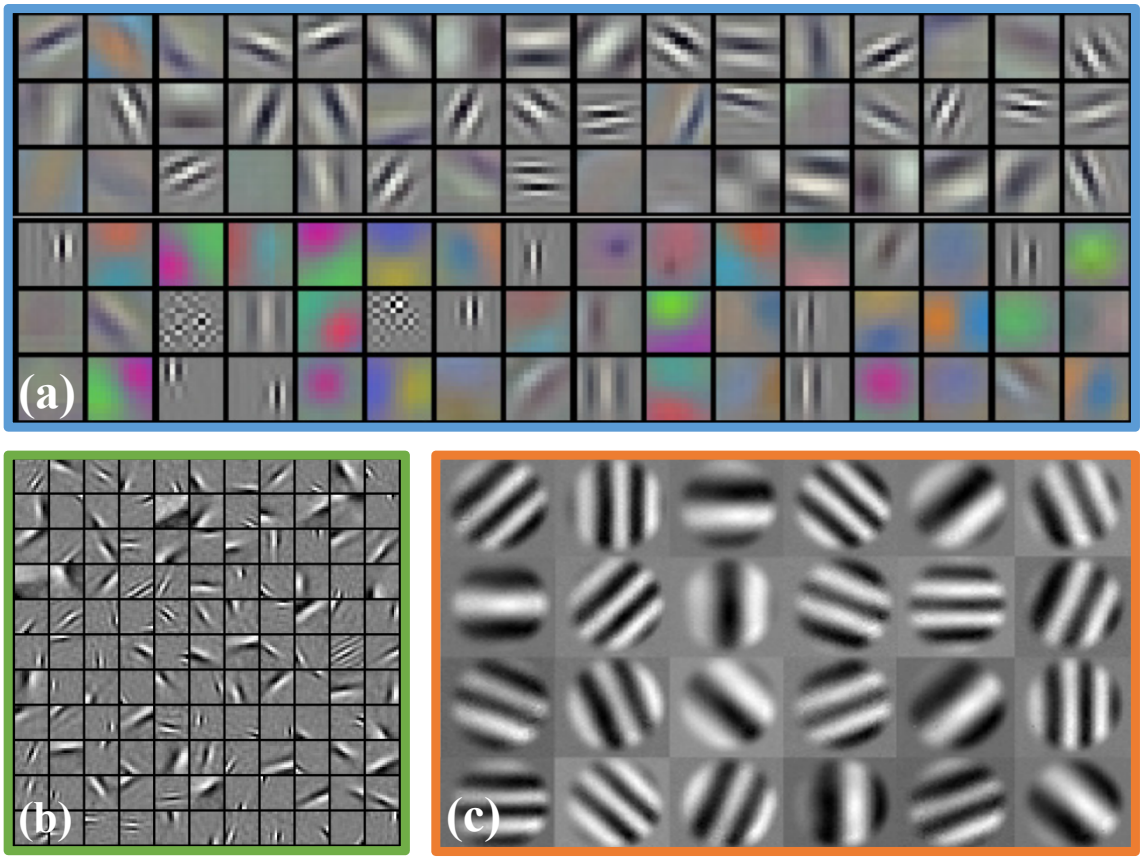


Figure 6.2: Comparison of first layer weights for three different systems: (a) Traditional deep learning architecture AlexNet trained on ImageNet [106], (b) Set of weights given by sparse coding constraints similar to the receptive fields of simple cells in the V1 [145], and (c) ASP optical impulse responses for 2D incidence angles [84].

6.3.1 Hardcoding the First Layer of CNNs

In partitioning a deep learning pipeline, a central question is what layers of the CNN should be implemented in hardware versus software. Hardcoded layers generally lead to significant energy savings provided a suitably efficient hardware implementation is used. However, maintaining the last layers of CNNs in software allows flexibility for network reconfiguration, transfer learning [147], and fine-tuning [14].

In our system, we are interested in optically computing the first layer of CNNs in hardware. We note recent research that shows the first layers of CNNs are application-independent and transferable [216]. Fine-tuning the CNN by retraining only the last few layers on a new application-domain leads to high accuracy. In particular, the first layer learned by most CNN architectures consists of oriented edge filters, color blobs, and color edges (as visualized AlexNet's [106] first layer in Figure 6.2(a)). These edge filters are not a surprise and are also found in the receptive fields of simple cells in the V1 layer of the human visual system. Olhausen and Field characterized these filters as Gabor wavelets, visualized in Figure 6.2(b), and showed how they perform sparse coding on natural image statistics [145].

Therefore hardcoding this first layer should be independent of application and roughly converges to the same set of Gabor filters for most networks. Our main idea is to use Angle Sensitive Pixels (ASPs) in our image sensor front end to compute this convolutional layer in the optical domain at low electronic power consumption.

6.3.2 Angle Sensitive Pixels

Background

ASPs are photodiodes, typically implemented in a CMOS fabrication process, with integrated diffraction gratings that image the Talbot diffraction pattern of incoming light [196]. These diffraction gratings give a sinusoidal response to incident angle of light given by the following equation [84]:

$$i(x,y) = 1 + m\cos\left(\beta\left(\cos\left(\gamma\right)\theta_x + \sin\left(\gamma\right)\theta_y\right) + \alpha\right), \tag{6.1}$$

where θ_x, θ_y are 2D incidence angles, α, β, γ are parameters of the ASP pixel corresponding to phase, angular frequency, and grating orientation, and m is the amplitude of the response. A tile of ASPs contain a diversity of angle responses, and are repeated periodically over the entire image sensor to obtain multiple measurements of the local light field. ASPs have been shown to capture 4D light fields [84] and polarization information [91]. An advantage of these sensors is that they are CMOS-compatible and thus can be manufactured in a low-cost industry fabrication process.

Optical Convolution

In particular, ASP responses to incidence angle allow optical convolution and edge filtering. Using two differential pixels of phase α and phase $\alpha + \pi$ (pixels A and B of Figure 6.3), we subtract their responses, $i_\alpha - i_{\alpha+\pi}$, to obtain the sinusoidal term of Equation 1 which depends solely on angle without the fixed DC offset. Figure 6.3 shows these measured differential pixel's impulse responses across an ASP tile. They resemble several different Gabor wavelets of different frequency, orientation, and phase which tile 2D frequency space. These impulse responses are convolved optically with objects in the scene during the capture process. The resulting ASP output correspond to edge filtered images as displayed in Figure 6.4. We use this optical convolution with Gabor wavelets to compute the first convolutional layer of a CNN.

Analogously, the V1 layer of the visual cortex contains Gabor wavelet responses for the receptive fields of simple cells, and Olhausen and Field showed that this representation is maximally efficient in terms of information transfer and sparse coding [145]. This is partly why we claim this system is bio-inspired: we are taking advantage of ASP's Gabor-like response to naturally compress the statistics of natural scenes to edges. This edge

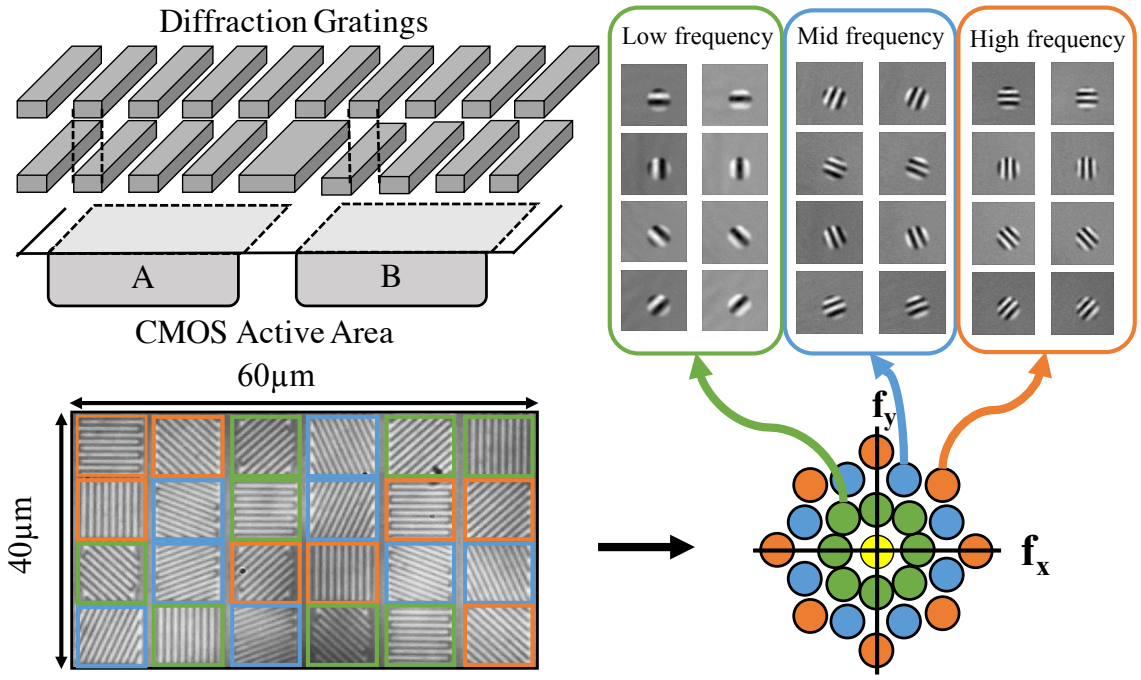


Figure 6.3: ASP Pixel Designs: ASP differential pixel design using diffraction gratings is shown. A 4×6 tile contains 10μm pixels whose optical responses are Gabor filters with different frequency, orientation, and phase. These filters act as bandpass filters in 2D frequency space [190].

representation has direct implications for the low power consumption of ASPs.

Energy and Bandwidth Efficiency

Prior work has designed ASP readout circuitry to leverage the sparseness of edge filtered images, enhancing energy efficiency [190]. Circuit readout that involves a differential amplifier can read out differential pixels, subtract their responses, and feed it to an analog-to-digital (ADC) converter [190]. This ADC is optimized to only convert pixels when there is sufficient edge information, leading to low power image sensing and digitization as compared to a traditional image sensor.

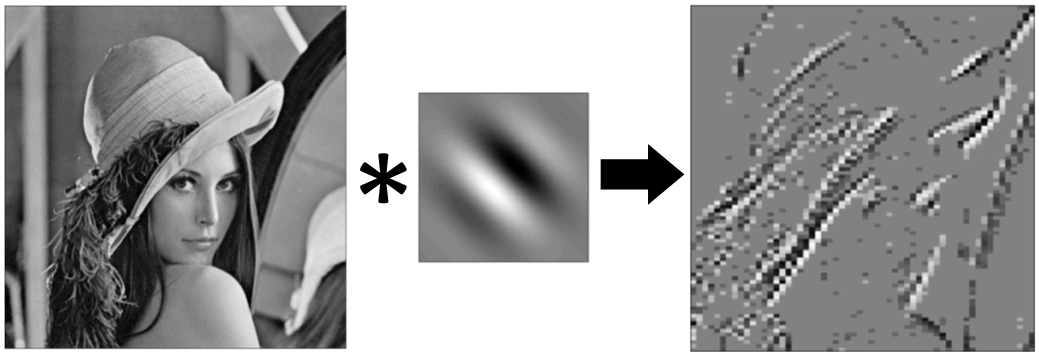


Figure 6.4: ASP Differential Output: Optical convolution of a scene with a differential ASP impulse response results in an edge filtered image (real images from prototype camera in [190]).

A comparison of an ASP-based image sensor [190] to a modern Sony mobile image sensor [172] is shown in Table 6.1. All numbers reported are from actual sensor measurements, but we caution the readers that these comparisons are approximate and do not take into account process technology and other second order effects. Note that while the current ASP sensor is lower power, it is also much smaller resolution than the Sony image sensor. However, we argue that regardless of the image sensor, the power savings of turning on the ADC to digitize only edges will always be advantageous for embedded vision.

Since edge data is significantly smaller to transmit, ASPs can also save on the bandwidth of image data sent off the image sensor, thus alleviating an I/O bottleneck from image sensor to CPU. Prior work has shown that ASPs obtain a **bandwidth reduction of 10:1 or 90%** for images by only storing non-zero coefficients of edges and using run-length encoding [190]. For a traditional image sensor, 1.2 Mbits is needed to digitize 150Kpixels (384 $\times$ 384) at 8 bit resolution while ASPs only require 120Kbits. We refer readers to [190] for more details about these circuit designs and their energy and bandwidth efficiency for ASP imaging.

	Sony (ISSCC 2015)	**ASP Image Sensor**
Resolution	5256 x 3934 (20M)	384 x 384 (effective ASP tile resolution: 96 x 64)
Energy consumption	Total power: 428 mW No breakdown of power reported	Total Power: 1.8 mW Pixel Array: 300 µW Amplifiers: 900 µW Timing/Addressing: 500 µW ADCs: 100 µW
Transmission bandwidth	Transmitting the entire image	Transmitting only edges
	1.2 Mbits/frame @ 384 ×384×8bits	120 Kbits/frame @ 384×384×8bits
	10:1 Compression ratio	
Capabilities	2D image and video capture	2D images and video, edge filtered images, light field information

Table 6.1: Comparison of ASP image sensor [190] and modern smartphone image sensor [172].

Limitations of ASPs for Visual Recognition

Some limitations with using ASPs for visual recognition include reduced image sensor resolution, low light efficiency, and depth-dependent edge filtering behavior. We outline these challenges and recent research to alleviate these issues.

Since a tile of ASPs is required to obtain different edge filters, image sensor resolution is reduced by the tile resolution. It is not clear how small ASP pixels can be fabricated, especially since a few periods of diffraction gratings are needed for adequate signal-to-noise ratio (SNR) and to avoid edge effects. However, recent research in interleaved photodiode design has increased pixel density by $2\times$ [168, 169]. Reduced resolution may have an adverse effect on vision tasks [39], although no critical minimum resolution/spatial frequency threshold has been suggested for image sensors to capture.

ASP pixels can suffer loss of light through the diffraction gratings as low as 10% relative quantum efficiency, which yields decreased SNR for differential edge responses. This in part explains the noisy visual artifacts present in the hardware prototype, and the need for large amounts of light in the scene. However, recent work in phase gratings [168, 169] have increased light efficiency up to 50% relative quantum efficiency.

Finally, the optical edge-filtering behavior of ASPs is depth-dependent since the optical responses only work away from the focal plane with a large aperture camera [196]. This depth-dependence limits the application of ASPs to wide aperture systems with shallow depth-of-field, but also enables the potential for depth and light field information to be utilized as scene priors (which we do not explore in the scope of this work).

6.4 Analysis

To analyze our proposed design and its tradeoffs, we developed a simulation framework to model both ASP image formation and CNNs. We simulate ASP image capture, and then propogate the resulting ASP edge images through the rest of the CNN. Typically this output data has dimensions W $\times$ H $\times$ D where there are D ASP filtered images, each of size W $\times$ H. We use the same input image resolution for both ASPs and baselines since we already accounted for image resolution in our normalized energy savings in Table 6.1.

For all our simulations, we use the ASP tile design of Figure 6.3 which matches the existing hardware prototype of [190]. We use 12 out of 24 of the ASP filters with cosine responses ($\alpha = 0$) and low, medium, and high angular frequencies. The other 12 filters have sine responses ($\alpha = \pi/2$) which did not yield suitably different convolution outputs, and thus these matching input channels caused gradient exploding and convergence issues. Finally, since our prototype ASP system does not have color pixels, we report all baselines with respect to grayscale for performance. All our dataset results are summarized in Figure 6.5 and discussed in the following subsection.

We use MatConvNet [185] to perform deep learning experiments and train on a NVIDIA GeForce GTX TITAN Black GPU.

6.4.1 Performance on Visual Recognition Tasks

We first analyze the performance of ASP Vision across several visual recognition tasks to show the broad applicability of this system. The datasets we benchmark include MNIST [114] for digit recognition, CIFAR-10/100 [105] for object recognition, and PF-

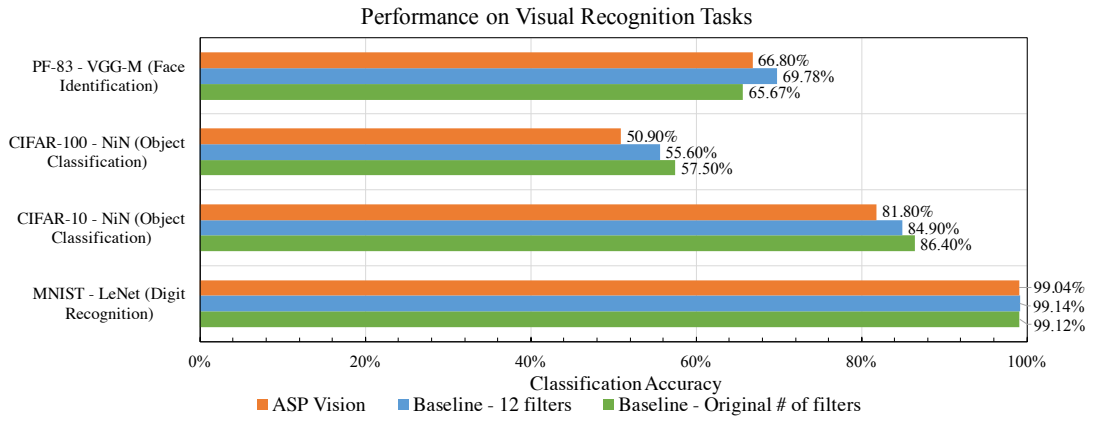


Figure 6.5: ASP Vision Performance: ASP Vision's performance on various visual recognition tasks, evaluated using three networks, LeNet [38], NiN [127] and VGG-M-128 [32], and over four different datasets: MNIST [114], CIFAR-10 [105], CIFAR-100 [105], and PF-83 [13].

	VGG-M		**NiN**		**LeNet**	
# of Conv. Layers	8		9		4	
Input Image Size	$224 \times 224 \times 3$		$32 \times 32 \times 3$		$28 \times 28 \times 1$	
# of First Layer Filters	96 (Original)	12 (Prototype)	192 (Original)	12 (Prototype)	20 (Original)	12 (Prototype)
First Layer Conv. Kernel	$7 \times 7 \times 96$	$7 \times 7 \times 12$	$5 \times 5 \times 192$	$5 \times 5 \times 12$	$5 \times 5 \times 20$	$5 \times 5 \times 12$
FLOPS of First Layer	708.0M	88.5 M	14.75M	921.6K	392 K	235 K
Total FLOPS	6.02G	3.83 G	200.3M	157 M	10.4 M	8.8 M
First Layer FLOPS Saving	11.76%	2.3%	7.36%	0.6%	3.77%	2.67%

Table 6.2: Network Structure and FLOPS: Common CNN architectures such as VGG-M-128 [32], NiN [127], LeNet [38] are compared for the FLOPS savings from optically computing the first layer of these networks. The actual FLOPS savings for the working prototype ASP Vision system are also included.

83 [13] for face identification.

For all experiments, we benchmark baselines with their original first layer number of filters (D) and also with $D = 12$ for a more fair comparison with ASP Vision when we analyze FLOPS in the next subsection.

MNIST: Our first simulation involved digit recognition on MNIST, 60,000 training and 10,000 test images of size 28×28. For a baseline, we use LeNet [38] which is a five layer CNN with both 20 and 12 first-layer filters to achieve 99.12%and 99.14% percent respectively. Using LeNet, ASP Vision achieved 99.04% performance.

CIFAR-10/100: Our second simulation involved the CIFAR-10/100 data sets [105] for object recognition with 50,000 training and 10,000 test images of size 32×32 (the 10/100 corresponds to the number of classes). Our baseline algorithm for these datasets was the Network in Network (NiN) structure [127] that uses CNNs with fully connected networks acting as inner layers. The baseline used both 92 and 12 first-layer filters to achieve respectively 86.40% and 84.90% percent on CIFAR-10, and 57.50% and 55.60% on CIFAR-100. Note again that these percentages are for grayscale images. ASP Vision achieved 81.8% and 50.9% respectively on CIFAR-10/100.

PF-83: Our final simulation on PF-83 [13] is an example of fine-grained classification to show that ASP features are transferable even for a difficult task like face identification (not to be confused with face verification or detection). The data consists of 13,002 images with size 256×256 with 83 classes of faces. Our baseline VGG-M-128 algorithm [32] achieved 65.67% and 69.78% percent on this data set with 192 and 12 first-layer filters respectively. Using ASP Vision, we achieved 66.8% percent on PF-83.

Across all datasets, ASP Vision was within 0.1-5.6% of the baseline accuracies. Note

that this comparable-to-slight degradation in performance comes with the energy savings of image sensing and transmission bandwidth by using ASPs.

6.4.2 FLOPS savings

The FLOPS saved by ASP Vision is dependent on both the network architecture and the size of the input images.

We first look at different CNN architectures and their potential savings from optically computing the first layer shown in Table 6.2. Additionally, since we simulate only our hardware prototype of 12 filters, we compare the FLOPS of our prototype ASP Vision system with those of modified CNNs with a 12-filter first layer. This comparison results in lower FLOPS savings, but yields higher visual recognition performance. Using an ASP with more numbers of filters would allow more FLOPS savings when compared to CNNs with the equivalent number of first-layer filters.

Secondly, FLOPS are input image size dependent as larger input image sizes will yield proportionally more FLOPS savings for an ASP Vision system. Even for a relatively deep network, the first layer still contributes a considerable amount of FLOPS if the input image is large. For example, the FLOPS of the first layer of GoogLeNet [174] is about 2.5% of the total FLOPS.

6.4.3 Noise analysis

In Figure 6.6, we simulate the effects of additive white noise during image sensing for MNIST images. We compare ASP Vision versus the baseline LeNet with SNR varying

from 9dB to 28dB. Note that at low SNRs, ASP Vision suffers more from accuracy degradation (9dB - 38.6%, 12dB - 77.9%) as compared to the baseline (9dB - 42.6%, 12dB - 83.6%). However, above 15dB SNR, both methods have high accuracy and are comparable.

6.4.4 ASP parameter design space

We finally explore how choice of ASP parameters affects performance with the salient parameters being angular frequency β and grating orientation χ. We performed a coarse sweep of $\beta \in [5, 50]$, $\chi \in [-\frac{\pi}{2}, \frac{\pi}{2})$ for one filter on MNIST, and found no strong dependence on parameters and performance.

We also ran sensitivity analysis on the parameter set by running 100 simulations using 6 randomized ASP filters each time on the MNIST dataset. We obtained a mean of 1.13% error with a standard deviation of 0.13%, which suggests there is no strong dependence of ASP parameters. This might be partly because the CNN learns to work with the filters it is given in the first layer.

6.5 Hardware Prototype and Experiments

Finally, to completely validate our system design, we show results of classification on an existing camera prototype for digit and face recognition. We report mean validation accuracy and standard deviation for 20 trials with a random split of 85% for training and 15% validation.

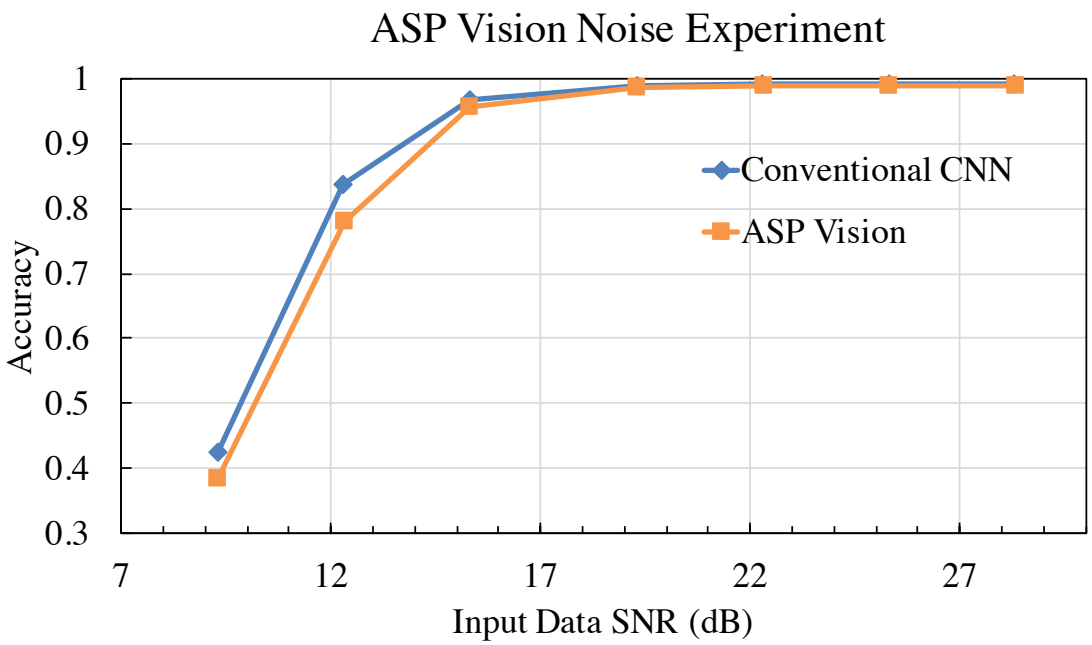


Figure 6.6: ASP Vision Noise Analysis: To explore the impact of noise to the performance of ASP Vision, we vary SNR from 9 dB to 28 dB and compared ASP Vision with baseline LeNet performance on MNIST.

The prototype camera system is the same setup as used in [84, 190]. A 5 mm $\times$ 5 mm CMOS image sensor was fabricated in a 180nm process, using a tile size of 4 $\times$ 6 ASPs with 10um pixels for a 64 $\times$ 96 resolution sensor. This sensor is placed behind a Nikon F1.2 lens for imaging small objects on an optical bench. See Figure 6.7 for picture of our prototype camera.

In general, our prototype camera suffers from high noise even after a fixed pattern noise subtraction. This may be due to noise issues from the readout circuits or even from external amplifiers on the printed circuit board. This limits the aesthetics of the ASP edge images, but we still achieved high accuracy in visual recognition. Further circuit design such as correlated double sampling and fabrication in an industrial CMOS image sensor process

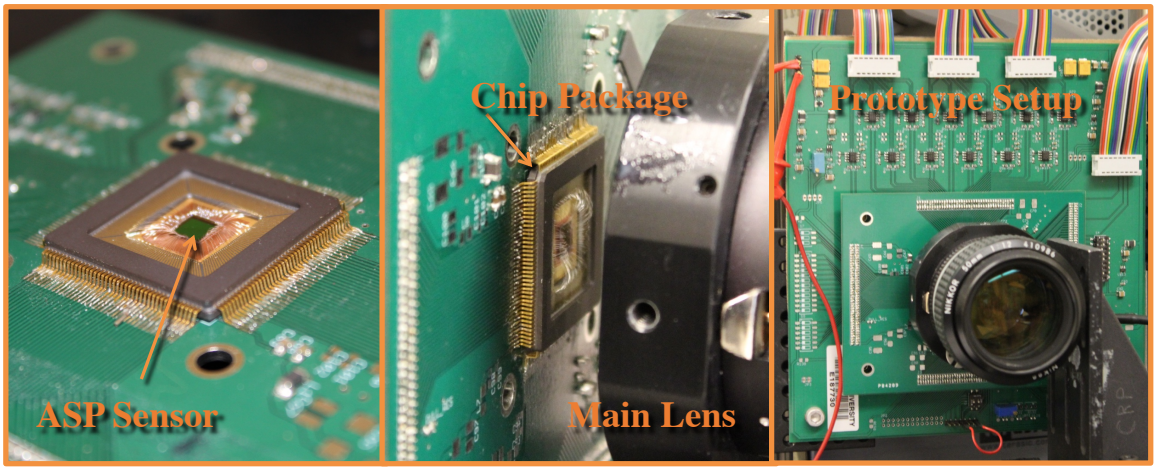


Figure 6.7: ASP Camera Setup: Working prototype with 5 mm x 5 mm CMOS ASP image sensor, F1.2 Nikon lens, and associated readout printed circuit board [84, 190].

could help alleviate these noise issues in the future.

Digit Recognition: Using a display with appropriate brightness approximately one meter away, we show images of the MNIST dataset, and capture ASP responses as shown in Figure 6.9. We captured over 300 pictures of real digits to be used in our learning experiment. We also used linear shifts and rotations to augment the size of our dataset to 2946 images. For real data, the baseline LeNet algorithm performed 91.26% with $\sigma = 2.77\%$ on the regular dataset, and 95.22% with $\sigma = 0.87\%$ on the augmented dataset. ASP Vision achieved 86.7% with $\sigma = 3.75\%$ on the regular dataset, and 94.61% with $\sigma = 0.68\%$ on the augmented dataset.

Face Identification: To test face identification, we took 200 pictures of 6 subjects approximately 2.5 meters away in the lab, and the edge responses and example results and errors are visualized in Figure 6.10. We used dataset augmentation again to increase the dataset to 7200 pictures. For the baseline NiN, we achieved 93.53% with $\sigma = 8.37\%$ on the regular dataset, and 94.73% with $\sigma = 4.2\%$ on the augmented dataset. ASP Vision

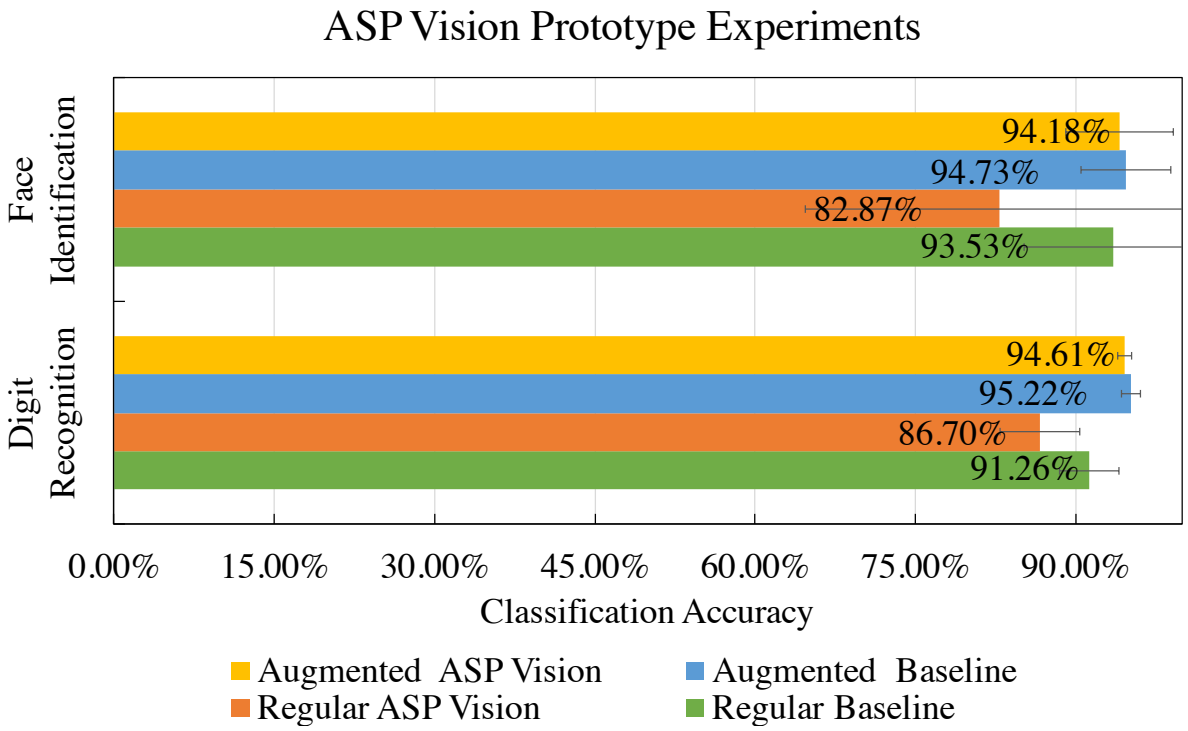


Figure 6.8: ASP Vision Prototype Experiments: Real-world digit recognition and face identification tasks were performed on ASP Vision prototype system. Accuracy and standard deviation for 20 trials are shown.

achieved 82.87% with $\sigma = 18.12\%$ on the regular dataset, and 94.18% with $\sigma = 5.04\%$ on the augmented dataset.

ASP Vision performs about 5-10% worse than baseline with regular data. After introducing linear shifts and rotations to augment the data, ASP Vision performs on par with conventional CNNs. These datasets may not generalizable and may exhibit underlying trends/bias due to the custom data acquisition. However, these results clearly show the feasibility of ASP Vision on a real working camera prototype.

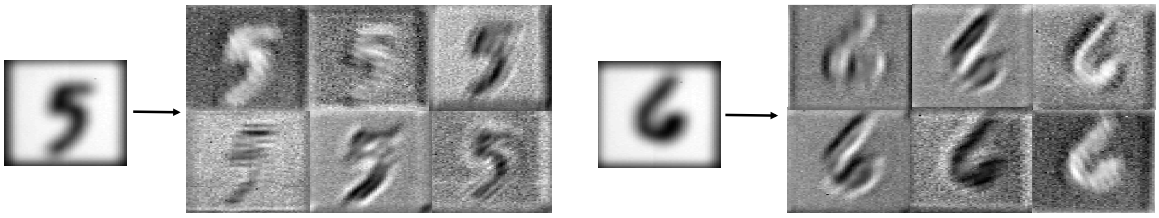


Figure 6.9: Digit Recognition: Digits are captured by the ASP image sensor, 6 of 12 sample edge responses from the tile are shown. ASP Vision achieved >90% accuracy in digit recognition on this dataset.

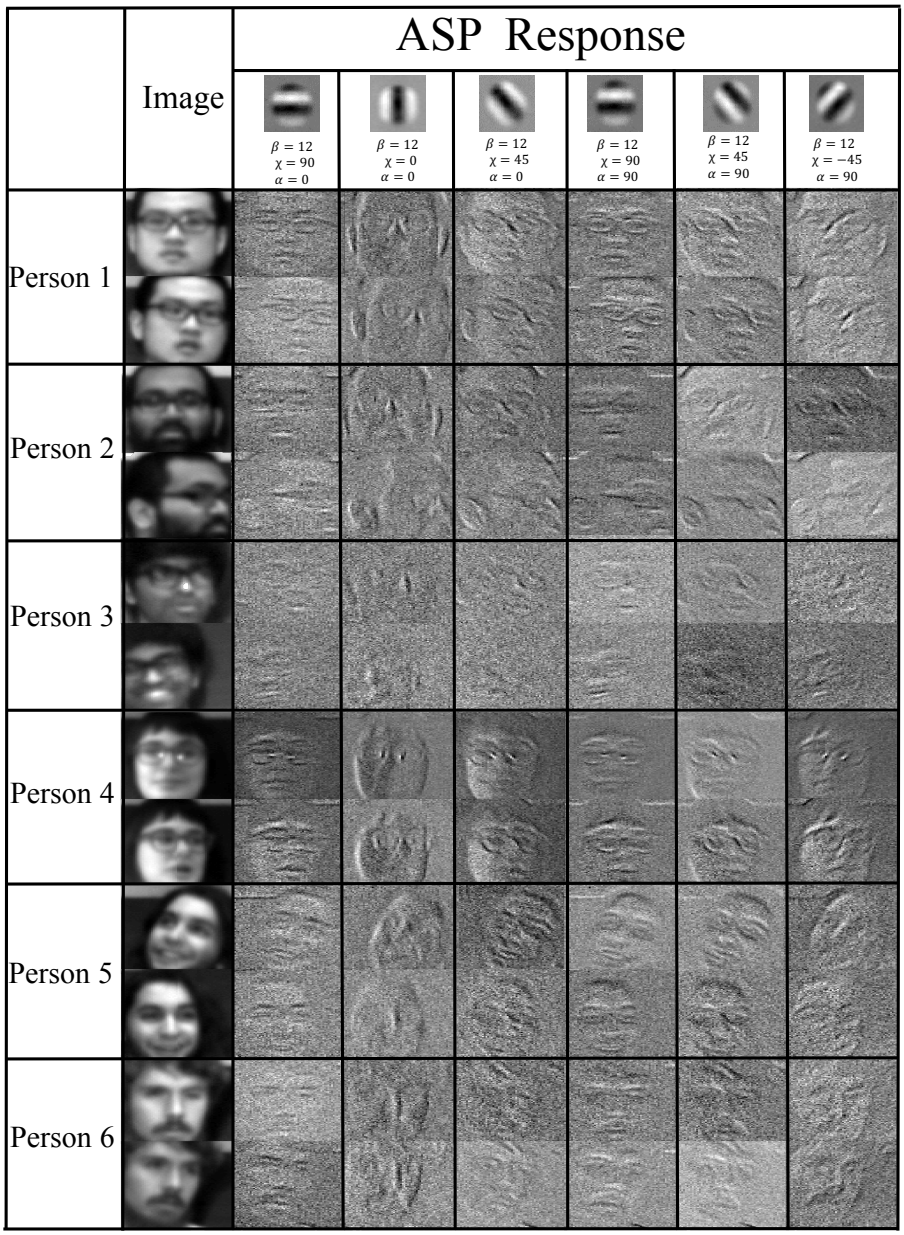
6.6 Discussion

Optically computing the first layers of CNNs is a technique that is not solely limited to ASPs. Sensors such as the DVS can compute edge features at low power/bandwidth [122], or using cameras with more general optical computation [220] could capture convolutional features. In addition, it is not possible to hardcode additional convolutional layers optically in ASPs beyond the first layer, limiting the potential energy savings. Fully optical systems for artificial neural networks using holography [50, 83, 152] or light waves in fiber [82] may achieve better energy savings.

We have presented an energy-efficient imaging system that uses custom Angle Sensitive Pixels for deep learning. We leverage energy savings and bandwidth reduction in ASPs while achieving good visual recognition performance on synthetic and real hardware data sets. We hope our work inspires sensor+ deep learning co-design for embedded vision tasks in the future.

6.7 Future work in plenoptic vision

We hope that this chapter inspires even further work in incoporating plenoptic information to visual recognition and other computer vision tasks. While we only use the edge filtering property of ASPs in this work, we could extend other ASP properties such as light field views/depth, polarization, or even optical flow as inputs to deep learning algorithms. Recently, researchers have used deep learning for improving the depth mapping for light fields [77] and even created datasets for light field material recognition [197]. Future work in plenoptic vision will help increase the usefulness of plenoptic cameras for modern imaging systems.



(a)

Image	ASP Response	False Guess
Person 3		Person 6
Person 5		Person 2
Person 6		Person 2

(b)

Figure 6.10: Face Recognition: 200 images of 6 subjects were captured in the lab. Edge responses for (a) correct and (b) misidentified identification is showed. ASP Vision achieved >90% accuracy

CHAPTER 7

CONCLUSION AND FUTURE DIRECTIONS

7.1 Summary

In this thesis, we have extended ASP imaging to capture additional plenoptic dimensions including angle and polarization, developed the framework for combining time-of-flight with plenoptic imaging, and used plenoptic information to improve the energy efficiency of convolutional neural networks performing visual recognition. This work has spanned multiple stacks of the software/hardware domain, from analog pixel design to board-level electronics and cameras all the way to high level machine learning and computer vision algorithms. We have shown a variety of new visual computing applications from our ASP setup, validating the benefits of experimental prototypes and real imaging data in the lab. In addition to research contributions in the main body of the thesis, we also present two chapters in the appendix: (1) a study of binary gradient cameras and deep learning, and (2) useful methodological work to help make computational CMOS sensors easier to design, simulate and test before being fabricated.

7.2 Limitations

With all the contributions we present, there are still limitations to ASP plenoptic imaging in general. By using one sensor for capturing multiple dimensions, we do suffer from reduced resolution such as the spatio-angular tradeoff in Ch. 3 or the lowered SNR of polarization sensing in Ch. 4. The Angle Sensitive Photogates proposed in Ch. 5 would

also suffer from low SNR due to loss of light from diffraction gratings coupled with photogate structures which suffer from modulation efficiency. Appendix B describes a new type of pixel amplifier designed to improve the SNR of ASP differential pixels in general, but this has not been tested in post-silicon yet. Realizing plenoptic image sensors that capture multiple dimensions at high sampling rates is still an open engineering challenge.

7.3 Future Research for ASP Imaging

Aside from the engineering/hardware research needed to improve ASPs noted in the previous section, there is also very interesting research avenues for ASP imaging. Wavelength is one plenoptic dimension that we do not address in this thesis that would be useful in biomedical applications. In addition, ASPs could capture extensions of the plenoptic function such as diffraction effects with Wigner functions [218, 31] or measuring coherency of light. Finally, active illumination and light transport parsing (see [146] for a broad overview) would be interesting to couple with ASPs to selectively allow certain plenoptic light paths to be captured at the sensor. This thesis aspires to be the starting point from which multimodal plenoptic imaging, both algorithms and new computational sensors, play a central role in the future of visual computing.

APPENDIX A

DEEP LEARNING USING ENERGY-EFFICIENT BINARY GRADIENT CAMERAS

This appendix presents some work on new types of computational cameras that output binary gradients, either in the temporal or spatial domain. We present a survey of deep learning applications for these cameras, and show that we can recover intensity information from binary spatial gradient images[1]. While this work is tangential to plenoptic imaging and ASPs, it does share commonalities in the use of computational sensors to save energy in modern vision pipelines.

A.1 Introduction

Recent advances in deep learning have significantly improved the accuracy of computer vision tasks such as visual recognition, object detection, segmentation, and others. Leveraging large datasets of RGB images and GPU computation, many of these algorithms now match, or even surpass, human performance. This accuracy increase makes it possible to deploy these computer vision algorithms in the wild. Power consumption, however, remains a critical factor for embedded and mobile applications, where battery life is a key design constraint.

For instance, Google Glass operating a modern face recognition algorithm has a battery life of less than 40 minutes, with image sensing and computation each consuming roughly 50% of the power budget [126]. Moreover, research in computer architecture has focused on energy-efficient accelerators for deep learning, which reduce the power footprint of

[1]This work was originally presented in S. Jayasuriya et al., "Deep Learning using Energy-efficient Binary Gradient Cameras" (submitted to CVPR 2017)

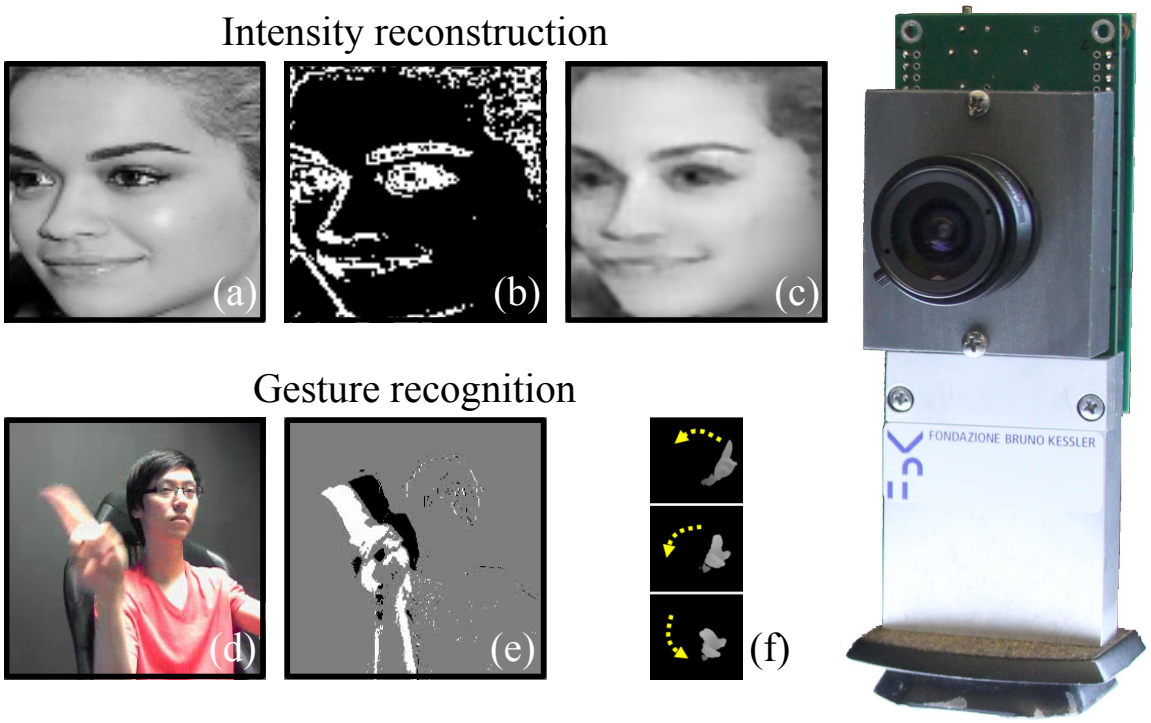


Figure A.1: Two of the tasks we study in the context of binary gradient images. Insets (a) and (d) are traditional pictures of the scene. Inset (b) is a simulated, spatial binary gradient, and (e) a simulated temporal binary gradient. From these we can reconstruct the original intensity image (c) or perform gesture recognition (f). We also used real data captured with the prototype shown on the right. Inset (f) is from Molchanov et al. [137].

neural network inference to the mW range [72, 28], bringing them in the same range of the power consumption as image sensing.

When the computer vision algorithms are too computationally intensive, or would require too much power for the embedded system to provide, the images can be uploaded to the cloud for off-line processing. However, even when using image or video compression, the communication cost can still be prohibitive for embedded systems, sometimes by several orders of magnitude [153]. Thus an image sensing strategy that reduces the amount of captured data can have an impact on the overall power consumption that extends beyond just acquisition and processing.

A large component of the image sensing power is burned to capture dense images or videos, meaning that each pixel is associated with a value of luminance, color component, depth, or other physical measurement. Not all pixels, however, carry valuable information: pixels capturing edges tend to be more informative than pixels in flat areas. Recently, novel sensors have been used to feed gradient data directly to the computer vision algorithms. [33, 199]. In addition, there has been a growing interested in event based cameras such as those proposed by Lichsteiner et al. [123]. These cameras consume significantly less power than traditional cameras, and record binary changes of illumination at the pixel level, and only output pixels when they become active. Another particularly interesting type of sensor was proposed by Gottardi et al. [66]. This sensor produces a binary image where only the pixels in high-gradient regions become active; depending on the modality of operation, only active pixels, or pixels that changed their activity status between consecutive frames, can then be read. The resulting images appear like binary edge images, see Figure A.2.

While these designs allow for a significant reduction of the power required to acquire, process, and transmit images, it also limits the information that can be extracted from

Figure A.2: A traditional image (left) and an example of real spatial binary gradient data (right). Note that these pictures were taken with different cameras and lenses and, thus, do not exactly match.

the scene. The question, then, becomes whether this results in a loss of accuracy for the computer vision algorithms, and if such loss is justified by the power saving.

A.1.1 Our Contributions

In this appendix, we focus on two aspects related to the use of binary gradient cameras for low-power, embedded computer vision applications.

First, we explore the tradeoff between energy and accuracy this type of data introduces on a number of computer vision tasks. To avoid having to hand-tune traditional computer vision algorithms to binary gradient data, we use deep learning approaches as benchmarks, and leverage the networks' ability to learn by example. We select a number of representative tasks, and analyze the change in accuracy of established neural network-based approaches, when they are applied to binarized gradients.

Second, we investigate whether the intensity information can be reconstructed from these images in post-processing, for those tasks where it would be useful for a human to visually inspect the captured image, such as long-term video surveillance on a limited

power budget. Unlike other types of gradient-based sensors, intensity reconstruction is an ill-posed problem for our type of data because both the direction and the sign of the gradient are lost, see Section A.5. To the best of our knowledge, in fact, we are the first to show intensity reconstruction from single-shot, spatial binary gradients.

We perform our formal tests simulating the output of the sensor on existing datasets, but we also validate our findings by capturing real data with the prototype developed by Gottardi et al. [66] and described in Section A.3.1.

We believe that this work presents a compelling reason for using binary gradient cameras in certain computer vision tasks, to reduce the power consumption of embedded systems.

A.2 Related Work

We describe the prior art in terms of the gradient cameras that have been proposed, and then in terms of computer vision algorithms developed for this type of data.

Gradient cameras can compute spatial gradients either in the optical domain [33, 220, 104], or on-board the image sensor, a technique known as focal plane processing [29, 117, 142, 75]. The gradients can be either calculated using adjacent pixels [66] or using current-mode image sensors [68]. Some cameras can also compute temporal gradient images, i.e. images where the active pixels indicate a temporal change in local contrast [66, 123]. Most of these gradient cameras have side benefits of fast frame rates and reduced data bandwidth/power due to the sparseness of gradients in a scene. In fact, the camera by Lichtsteiner et al. can read individual pixels when they become active [123]. Moreover, the

fact that gradient cameras output a function of the difference of two or more pixels, rather than the pixel values themselves, allows them to deal with high-dynamic-range scenes.

Applications of gradient cameras were first exposited in the work by Tumblin et al., who described the advantages of reading pixel differences rather than absolute values [182]. A particular area of interest for temporal binary gradients and event-based cameras is SLAM (simultaneous localization and mapping) and intensity reconstruction. Researchers have shown SLAM [200], simultaneous intensity reconstruction and object tracking [100], combined optical flow and intensity reconstruction [11], and simultaneous depth, localization, and intensity reconstruction [101]. In addition, some early work has focused on using spiking neural networks for event-based cameras [143]. The common denominator to all of these techniques is that the camera, or at least the scene, must be dynamic: the sensor does not output any information otherwise. For tradeoffs between energy and visual recognition accuracy, recent work proposed optically computing the first layer of convolutional neural networks using Angle Sensitive Pixels [33]. However, the camera required slightly out-of-focus scenes to perform this optical convolution and did not work with binary gradient images.

In our work, we focus on the camera proposed by Gottardi et al. [66], which can produce spatial binary gradients, and can image static scenes as well as dynamic ones. Gasparini et al. showed that this camera can be used as a long-lifetime node in wireless networks [56]. This camera was also used to implement low-power people counter [55], but only in the temporal gradient modality (see Section A.3.1).

A.3 Binary Gradient Cameras

In this section, we define the types of binary gradient images we are considering and we analyze the power and high dynamic range benefits from such cameras.

A.3.1 Operation

For spatial binary gradients, we refer to cameras where a pixel becomes active when a local measure of contrast is above threshold. Specifically, for two pixels i and j, we define the difference $\Delta_{i,j} = |I_i - I_j|$, where I is the measured pixel's brightness. We also define a neighborhood ν consisting of pixel P and the pixels to its left, L, and top, T. The output at pixel P will then be:

$$G_S(\mathrm{P}) = \begin{cases} 1 & \text{if } \max\limits_{i,j \in \nu} \Delta_{i,j} > T \\ 0 & \text{otherwise} \end{cases}, \tag{A.1}$$

where T is a threshold set at capture time. The output of this operation is a binary image where changes in local spatial contrast above threshold yield a 1, else a 0, see Figure A.2. Note that this operation is an approximation of a binary local derivative: $\Delta_{T,L}$ alone can trigger an activation for P, even though the intensity at P is not significantly different from either of the neighbors'. It can be shown that the consequence of this approximation is a "fattening" of the image edges by a factor of roughly $\sqrt{2}$ when compared to the magnitude of the a gradient computed with regular finite differences. The advantage of this formulation is that it can be implemented efficiently in hardware.

For temporal binary gradients, the sensor proposed by Lichtsteiner et al. [123], which works asynchronously, outputs +1 (-1) for a pixel whose intensity increases (decreases) by

a certain threshold, and 0 otherwise. The sensor proposed by Gottardi et al. produces a slightly different image for temporal gradients, where the value of a pixel is the difference between its current and previous binary spatial gradient [66]:

$$G_T(\mathrm{P}, t) = \max\left(0, |G_S(\mathrm{P}, t) - G_S(\mathrm{P}, t-1)|\right), \tag{A.2}$$

where we made the dependency on time t explicit. This is implemented by storing the previous value in a 1-bit memory collocated with the pixel to avoid unnecessary data transfer. An image produced by this modality can be seen in Figure A.1(e).

A.3.2 Power Considerations

Binary gradient cameras have numerous advantages in terms of power and bandwidth. A major source of power consumption in modern camera sensors is the analog-to-digital conversion and the transfer of the 12-16 bits data off-chip, to subsequent image processing stages. Gradients that employ 1 or 2 bits can significantly reduce both the cost for the conversion, and the amount data to be encoded at the periphery of the array. In fact, the sensor only transfers the addresses of the pixels that are active, and when no pixels are active, no power is used for transferring data.

Comparing power consumption for sensors of different size, technology, and mode of operation is not easy. Our task is further complicated by the fact that the power consumption for a binary gradient sensor is a function of the contrast in the scene. However, here we make some assumptions to get a very rough figure. Gottardi et al. [66] report that the number of active pixels is usually below 25% (in the data we captured, we actually measured that slightly less than 10% of the pixels were active on average). The power consumption for the sensor by Gottardi et al. can be approximated by the sum of two components. The

first, independent of the actual number of active pixels, is the power required to scan the sensor and amounts to 0.0024μW/pixel. The second is the power required to deliver the addresses of the active pixels, and is 0.0195μW/pixel [54]. At 30fps, this power corresponds to 7.3pJ/pixels. A modern image sensor, for comparison, is over 300pJ/pixel [84]. Once again, these numbers are to be taken as rough estimates.

A.4 Experiments

In this section, we describe the vision tasks we used to benchmark spatial and temporal binary gradients. For the benchmarks involving static scenes or single images, we could only test spatial gradients. We used TensorFlow and Keras to construct our networks. All experiments were performed on a cluster of GPUs with NVIDIA Titan X's or K80s. For all the experiments in this section, we picked a reference baseline network appropriate for the task, we trained it on intensity or RGB images, and compared the performance of the same architecture on data that simulates the sensor by Gottardi et al. [66]. An example of such data can be seen in Figure A.1(b) and A.1(c). Table A.1 summarizes all the comparisons we describe below.

A.4.1 Computer Vision Benchmarks

Object Recognition — We used MNIST [114] and CIFAR-10 [105] to act as common baselines, and for easy comparison with other deep learning architectures, on object recognition tasks. MNIST comprises 60,000, 28x28 images of handwritten digits. CIFAR-10 has 60,000, 32x32 images of objects from 10 classes, with 10,000 additional images for

Task	Dataset	Traditional	Binary gradient
Recognition	MNIST [114]	99.19%	98.43%
	CIFAR-10 [105]	77.01%	65.68%
	NVGesture [137]	72.5%	G_T: 74.79% G_S: 65.42%
Head pose	300VW [165]	0.6°	1.8°
	BIWI Face Dataset [48]	3.5°	4.3°
Face detection — WIDER [211]	Easy	89.2%	74.5%
	Medium	79.2%	60.5%
	Hard	40.2%	28.3%

Table A.1: Summary of the comparison between traditional images and binary gradient images on visual recognition tasks.

validation. For these tasks we used LeNet [38].

On MNIST, using simulated binary gradient data degrades the accuracy by a mere 0.76%. For CIFAR-10, we trained the baseline on RGB images. The same network, trained on the simulated data, achieves a loss in accuracy of 11.33%. For reference, using grayscale instead of RGB images causes a loss of accuracy of 4.86%, which is roughly comparable to the difference in accuracy between using grayscale and gradient images—but without the corresponding power saving.

Head Pose Regression — We also explored single-shot head pose regression, an important use-case for human-computer interaction, and driver monitoring in vehicles. We used two datasets to benchmark the performance of gradient cameras on head pose regression. The first, the BIWI face dataset, contains 15,000 images of 20 subjects, each accompanied by a depth image, as well as the head 3D location and orientation [48]. The second, the 300VW dataset, is a collection of 300 videos of faces annotated with 68 landmark points [165]. We used the landmark points to estimate the head orientation.

On the BIWI dataset, training a LeNet from scratch did not yield network convergence.

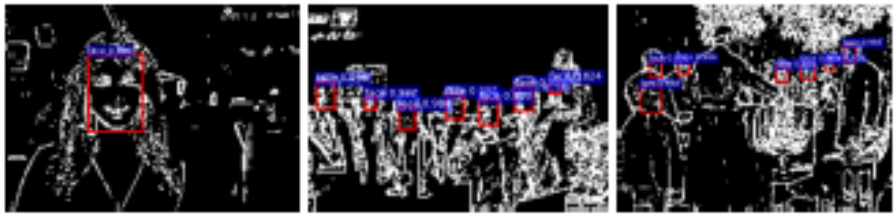


Figure A.3: Face detection on binary spatial gradient images simulated from the WIDER dataset.

Therefore, we used a pretrained VGG16 [167] network on the RGB images. We then fine-tuned the network on the simulated binary gradient data. The network trained on simulated binary gradient data yields a degradation of estimation accuracy of a 0.8 mean degree error per pixel. On the 300VW dataset, we trained LeNet on the simulated data. The mean angular error increases by 1.2 degrees per pixel, which is small when accounting for the corresponding power saving.

Face Detection — Another traditional vision task is face detection. For this task we trained the network on the WIDER face dataset, a collection of 30,000+ images with 390,000+ faces, and is organized in three categories for face detection: easy, medium, and hard [211]. Figure A.3 shows representative images of different levels of difficulty. Note that this dataset is designed to be very challenging, and includes pictures taken under extreme scale, illumination, pose, and expression changes, among other factors.

For this task, we used the network proposed by Ren et al. [155]. Once again, we trained it on both the RGB and the simulated binary gradient images. The results are summarized in Table A.1. On this task, the loss in accuracy due to using the binary gradient data is more significant, ranging from 11.9% to 18.7%, depending on the category.

Gesture Recognition — Our final task was gesture recognition. Unlike the previous benchmarks, whose task can be defined on a single image, this task has an intrinsic temporal

component: the same hand position can be found in a frame extracted from two different gestures. Therefore, for this task we test both the spatial and temporal modalities.

We used the dataset released by Molchanov et al., which contains 1,500+ hand gestures from 25 gesture classes, performed by 20 different subjects [137]. The dataset offers several acquisition modalities, including RGB, IR, and depth, and was randomly split between training (70%) and testing (30%) by the authors. The network for this algorithm was based on [137], which used an RNN on top of 3D convolutional features. We limited our tests to RGB inputs, and did not consider the other types of data the dataset offers, see Figure A.4. As shown in Table A.1, the simulated spatial binary gradient modality results in an accuracy degradation of 7.08% relative to RGB images and 5.41% relative to grayscale. However, as mentioned before, this task has a strong temporal component and one would expect that the temporal gradient input should perform better. Indeed, the temporal modality yields increased accuracy on both grayscale (+3.96%) and RGB (+2.29%) data. This is a significant result, because the additional accuracy is possible thanks to data that is actually cheaper to acquire from a power consumption standpoint. Note that the input to the network is a set of non-overlapping clips of 8 frames each, so the network can still "see" temporal information in modalities other than the temporal binary gradients.

Across a variety of tasks, we see that the accuracy on binary gradient information varies. It is sometimes comparable to, and sometimes better than, the accuracy obtained on traditional intensity data. Other times there is a significant accuracy loss is significant, as is the case with face detection. We think that this is due in part to the task, which can benefit from information that is lost in the binary gradient data, and in part to the challenging nature of the dataset. Our investigation suggests that the choice of whether a binary gradient camera can be used to replace a traditional sensor, should account for the task at hand and its accuracy constraints. Note that we did not investigate architectures that may better fit

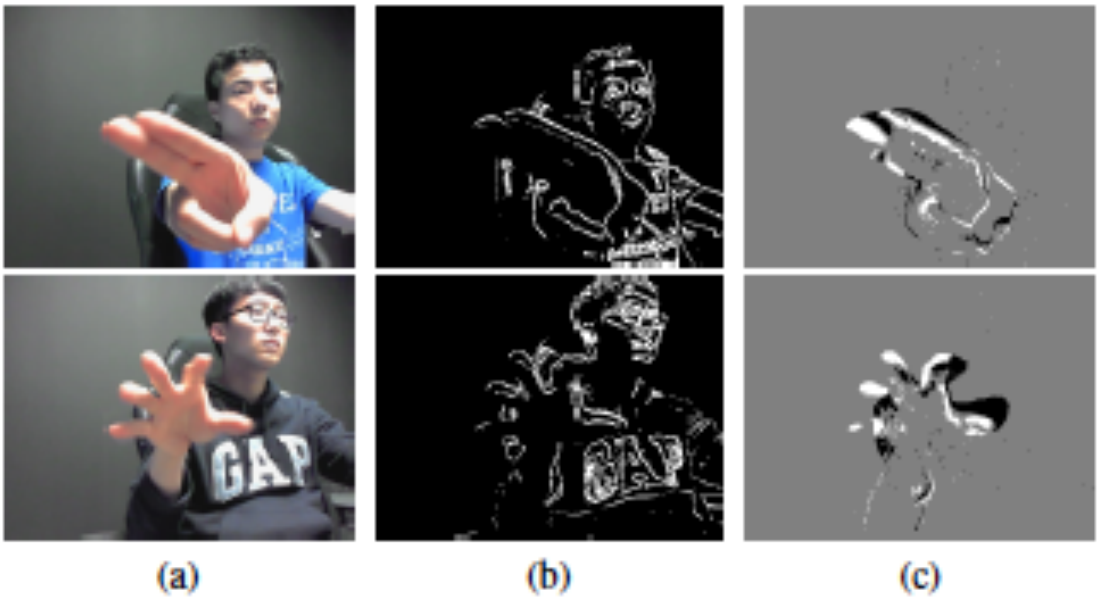


Figure A.4: Two frames from the NVIDIA Dynamic Hand Gesture Dataset [137], (a), the corresponding spatial binary gradients, (b), and temporal binary gradients, (c).

this type of data, and which may have an impact on accuracy. We leave the investigation for future work, see also Section A.7.

A.4.2 Effects of Gradient Quantization

In this section, we study the tradeoff between power consumption and accuracy of binary gradient cameras. One factor that has a strong impact on both, is the number of bits we use to quantize the gradient, which, so far, we have assumed to be binary. Designing a sensor with a variable number of quantization bits, while allowing for low power consumption, could be challenging. However, graylevel information can be extracted from a binary gradient camera by accumulating multiple frames, captured at a high frame rate, and by combining them into a sum weighted by the time of activation [54]. For the sen-

sor proposed by Gottardi et al. [66], the power of computing this multi-bit gradient can be estimated as:

$$P = 2^N \cdot P_{\text{scan}} + P_{\text{deliver}}, \tag{A.3}$$

where N is the number of quantization levels, P_{scan} is the power required to scan all the rows of the sensor, and P_{deliver} is the power to deliver the data out of the sensor, which depends on the number of active pixels [54]. Despite the fact that P_{deliver} is an order of magnitude larger than P_{scan}, Equation A.3 shows that the total power quickly grows with the number of bits.

To study the compromise between power and number of bits, we simulated a multi-bit gradient sweep on CIFAR-10, and used Equation A.3 to estimate the corresponding power consumption. Figure A.5 shows that going from a binary gradient to an 8-bit gradient allows for a 3.89% increase in accuracy, but requires more than 80 times the power. However, a 4-bit gradient may offer a good compromise, seeing that it only requires 7% of the power needed to estimate an 8-bit gradient (6 times the power required for the binary gradient), at a cost of only 0.34% loss of accuracy. This experiment points to the fact that the trade-off between power consumption and accuracy can be tuned based on the requirements of the task, and possibly the use-case itself. Moreover, because in the modality described above N can be changed at runtime, one can also devise a strategy where the quantization levels are kept low in some baseline operation mode, and increased when an event triggers the need for higher accuracy.

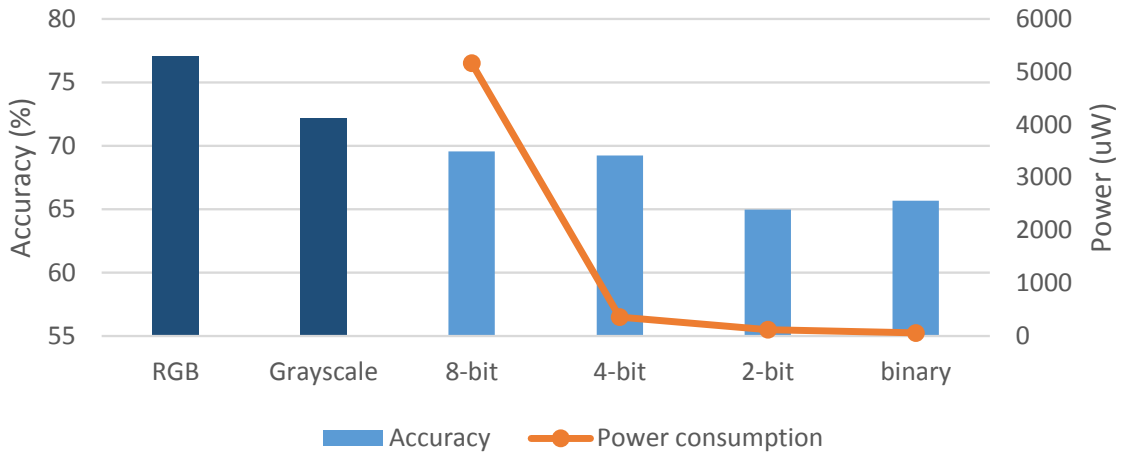


Figure A.5: Quantization vs power consumption vs accuracy tradeoff on CIFAR-10. Note the significant drop in power consumption between 8 and 4 bits, which is not reflected by a proportional loss of accuracy, see Section A.4.2.

A.5 Recovering Intensity Information from Spatial Binary Gradients

In addition to the automated computer vision machinery, some applications may require a human observer to look at the data. An example is video surveillance: a low-power automatic system can run continuously to detect, for instance, a person coming in the field of view. When such an event is detected, it may be useful to have access to intensity data, which is more easily accessible by a human observer. One solution could be that a more power-hungry sensor, such as an intensity camera is activated when the binary gradient camera detects an interesting event [71]. Another solution could be to attempt to recover the grayscale information from the binary data itself. In this section, we show that this is indeed possible.

We outlined previous work on intensity reconstruction from temporal gradients in Section 3.2. Currently available techniques, such as the method by Bardow et al. [11], use advanced optimization algorithms and perform a type of Poisson surface integration [3]

to recover the intensity information. However, they focus on the temporal version of the gradients. As a consequence, these methods can only reconstruct images captured by a moving camera, which severely limits their applicability to real-world scenarios.

To the best of our knowledge, there has been no work on reconstructing intensity images from a single binary spatial gradients image, in part because this problem does not have a unique solution. Capturing a dark ball against a bright background, for instance, would yield the same exact binary spatial gradient as a bright ball on a dark background. This ambiguity prevents the methods of surface integration from working, even with known or estimated boundary conditions.

We take a deep learning approach to intensity reconstruction, so as to leverage the network's ability to learn priors about the data. For this purpose, we focus on the problem of intensity recovery from spatial gradients of faces. While we cannot hope to reconstruct the exact intensity variations of a face, we aim to reconstruct facial features from edge maps so that it can be visually interpreted by a human. Here we describe the network architecture we propose to use, and the synthetic data we used to train it. In Section A.6 we show reconstructions from real data we captured using a binary gradient camera prototype.

Our network is inspired by the autoencoder architecture recently proposed by Mao et al. [133]. The encoding part consists of 5 units, each consisting of two convolutional layers with leaky ReLU nonlinearities followed by a max pooling layer. The decoding part is symmetric, with 5 units consisting of upsampling, a merging layer for skip connections that combines the activations after the convolutions from the corresponding encoder unit, and two convolutional layers. See Figure A.6 for our network structure. We trained this architecture on the BIWI and WIDER datasets.

For the BIWI dataset, we removed two subjects completely to be used for testing. Fig-

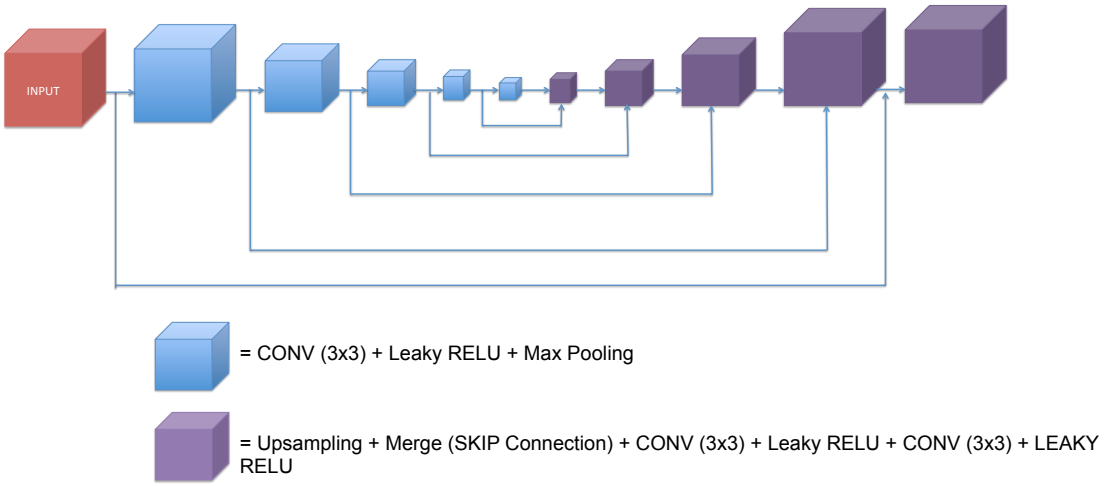


Figure A.6: The architecture of the autoencoder used to reconstruct intensity information from spatial binary gradient images.

ure A.7 shows an embedded animation of the two testing subjects. As mentioned above, the solution is not unique given the binarized nature of the gradient image, and indeed the network fails to estimate the shade of the first subject's sweater. Nevertheless, the quality is sufficient to identify the person in the picture, which is surprising, given the sparseness of the input data.

The WIDER dataset, does not contain repeated images of any one person, which guarantees that no test face is seen by the network during training. We extracted face crops by running the face detection algorithm described in Section A.4.1, and resized them to 96x96, by either downsampling or upsampling, unless the original size was too small. Figure A.8 shows some results of the reconstruction. Note that the failure cases are those where the quality of the gradients is not sufficient (Figure A.8(i)), or the face is occluded (Figure A.8(j)). The rest of the faces are reconstructed unexpectedly well, given the input. Even for the face in Figure A.8(j) the network is able to reconstruct the heavy makeup reasonably well.

Figure A.7: intensity reconstruction (middle pane) on the binary data (left pane) simulated from the BIWI dataset [48]. The ground truth is on the right.

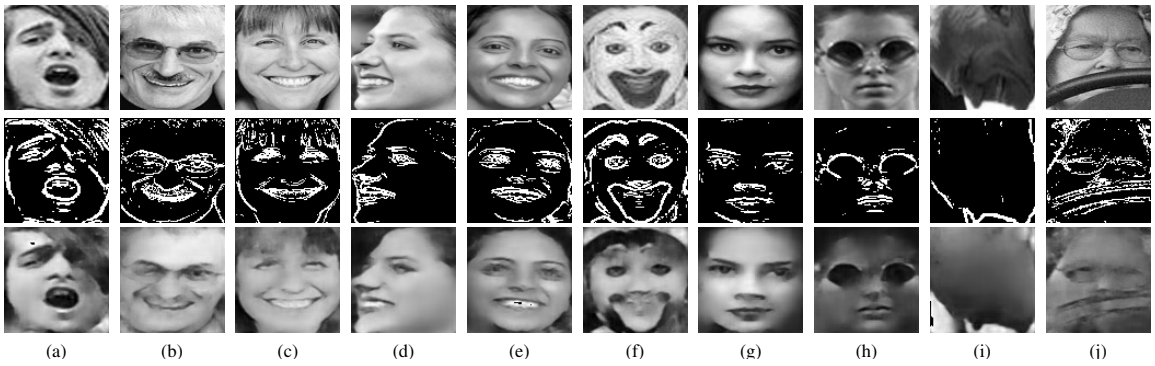


Figure A.8: Intensity reconstruction (bottom row) on the binary data (middle row) simulated from the WIDER dataset [211]. The ground truth is in the top row. Note that our neural network is able to recover the fine details needed to identify the subjects. We observed that failure cases happen when the gradients are simply too poor (i) or the face is occluded (j).

A.6 Experiments with a Prototype Spatial Binary Gradient Camera

In this section we validate our findings by running experiments directly on real binary gradient images. As a reminder, all the comparisons and tests we described so far were performed on data obtained by simulating the behavior of the binary gradient camera. Specifically, we based our simulator on Equation A.1, and tuned the threshold T to roughly match the appearance of the simulated and real data, which we captured with the prototype camera described by Gottardi et al. [66]. At capture time, we use the widest aperture setting possible to gain the most light, though at the cost of a shallower depth of field, which we did not find to affect the quality of the gradient image. We also captured a few grayscale images of the same scene with a second camera set up to roughly match the field of views of the two. Figure A.2, shows a comparison between a grayscale image and the (roughly) corresponding frame from the prototype camera. Barring resolution issues, at visual inspection we believe our simulations match the real data.

A.6.1 Computer Vision Tasks on Real Data

To qualitatively validate the results of our deep learning experiments, we ran face detection on binary gradient data captured in both outdoor and indoor environment. We could not train a network from scratch, due to the lack of a large dataset, which we could not capture with the current prototype—and the lack of ground truth data would have made it impossible to measure performance quantitatively anyway. We trained the network described in Section A.4.1 on simulated data resized to match the size of images produced by the camera prototype, and then we directly ran inference on the real data.We found that the same network worked well on the indoor scenes, missing a small fraction of the faces, and

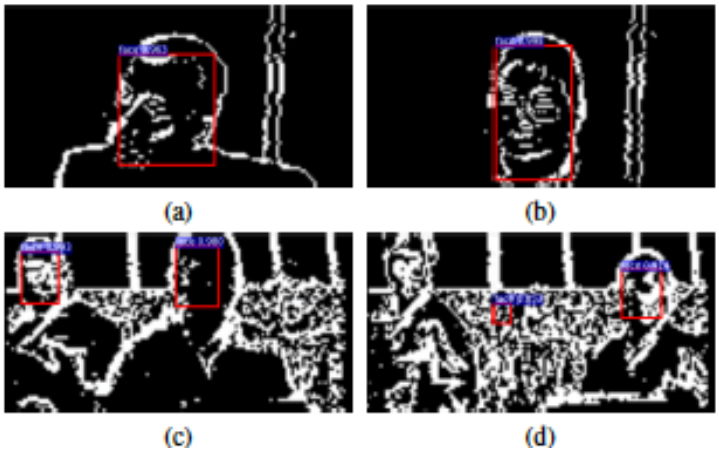


Figure A.9: Face detection task on spatial gradient images captured with the camera prototype. The top and bottom rows show frames from an indoor and an outdoor sequence, respectively. The misdetection rate is significantly higher in outdoor sequences, as seen in insect (d).

typically those whose pose deviated significantly from facing forward. On the other hand, the network struggled more when dealing with the cluttered background typical of the outdoor setting, where it missed a significant amount of faces. We ascribe this issue to the low spatial resolution offered by the prototype camera, which is only 128x64 pixels. However, this is not a fundamental limitation of the technology, and thus we expect it to be addressed in future versions. Figure A.9 shows a few detection results for both environment.

A.6.2 Intensity Reconstruction on Real Data

Another qualitative validation we performed was intensity reconstruction from data captured directly with the camera prototype. We trained the network on synthetic data generated from the WIDER dataset, and performed forward inference on the real data. Once again, we could not perform fine-tuning due to the lack of ground truth data—the data from an intensity camera captured from a slightly different position, and with different

lenses, did not generalize well. While the quality of the reconstruction is slightly degraded with respect to that of the synthetic data, the faces are reconstructed well. See FigureA.10 for a few example. Note that despite the low resolution (these crops are 1.5 times smaller than those in Figure A.8), the face features are still distinguishable.

Remember that here we are reconstructing intensity information from a single frame: we are not enforcing temporal consistency, nor we use information from multiple frames to better infer intensity. We find that the quality of the reconstruction of any single frame varies: some reconstructions from real data allow the viewer to determine the identity of the subject, others are more similar to average faces.

Figure A.10: Intensity reconstruction result inferred by the network described in Section A.5 and trained on the WIDER simulated data. The top row shows 64x64 face crops captured with the prototype camera, the bottom the corresponding reconstructed images. While the quality is not quite on par with the intensity reconstructions, it has to be noted that the resolution of the crops in Figure A.8, is 96x96, i.e. 1.5x larger.

A.7 Discussion

To further decrease the power consumption in computer vision tasks, we could couple binary gradient images with binary neural networks. Recently, new architectures have been proposed that are use elementary layers (convolutions, fully connected layers) using binary weights, yielding an additional 40% in power savings in computation [36]. We evaluated these binary neural networks (BNNs) on, MNIST, CIFAR-10, and SVHN [140]. (The latter is a dataset of $\sim$100K house street numbers.) On MNIST, a 1.57% error on gradient images increased to 2.23 % error by employing a BNN. For CIFAR-10, a 11% error on gradient images increased to 30% with the BNN. Finally for SVHN, a 3% error on binary gradient images increased to 12% with the BNN. Thus, while there are considerable power savings from using a BNN, it is still an open question of how to couple these networks with binary gradient data from novel sensors. We leave this as an avenue for future work on end-to-end binary vision systems.

We have conducted a thorough exploration of different computer vision tasks that can leverage binary gradient images. Certain tasks, such as object recognition and face detection, suffer more degradation in accuracy. Other tasks, such a gesture recognition, see an increase in accuracy. All with a significant power saving. In addition, we propose to use an autoencoder network to learn the prior distribution of a specific class of images to solve the under-constrained problem of recovering intensity information from binary spatial edges.

APPENDIX B

DIGITAL HARDWARE-DESIGN TOOLS FOR COMPUTATIONAL SENSOR DESIGN

B.1 Overview

As noted in Chapter 5, there is a large engineering effort that lies between the conceptual design of new image sensors and their fabrication in CMOS technology, not to mention the resulting experimental characterization and testing after the silicon chips return from fabrication. Industry workflows typically involve hundreds of engineers whose work spans analog circuit design, computer architecture, digital VLSI, physical layout, power routing and clock timing to name a few. While Moore's law has allowed more computing power on the same area of silicon, it has also caused the complexity of design to scale exponentially. This design complexity and the amount of infrastructure necessary to tapeout mixed-signal image sensors is a **major bottleneck** for compuational imaging researchers in academia.

It is the goal of this appendix to provide what we believe is the first workflow[1] for designing computational image sensors that is accessible for academic institutions. We do note that we still require industry-level tools and software packages, so this workflow does not reduce the cost of tapeing out an image sensor. However, we emulate the computer architecture research community by leveraging simulation to rapidly prototype ideas in co-designing image sensors with embedded DSP blocks. This incremental design process can yield valuable feedback to researchers, and the resulting design can be mostly automated for reduced effort in physical chip layout for CMOS fabrication. While none

[1]Some of this work is originally presented in C. Torng et al, "Experiences using a Novel Python-Based Hardware Modeling Framework for Computer Architecture Test Chips", Poster at HOT CHIPS 2016 [35].

of the individual tools used are novel from a research perspective, we still believe that this methodological work enables faster research prototyping and vertical integration amongst the hardware/software stack.

We leverage hardware design tools such as PyMTL [130] and Verilog RTL to quickly design computational signal processing blocks and control circuitry for image sensors. These digital designs can be synthesized with either a standard cell library provided by a process design kit (PDK) or we show how to design and extend a custom standard cell library. After synthesis, we then place-and-route the digital cells, manage power and timing, and can import the resulting digital blocks back to interface with custom analog circuit blocks (such as image sensors and analog amplifiers). To validate this workflow, we detail two case studies: (1) the design of a pipelined 32-bit RISC microprocessor with on-chip 16KB SRAM memory and C-to-RTL high level synthesis (HLS) digital accelerators, and (2) Angle Sensitive Photogates for depth field imaging pixels controlled by a high dynamic range amplifier and digitally-synthesized control logic as well as a 8 bit fixed-point divider.

B.2 Design Flow

In this section, we describe our workflow for using hardware-design tools to enable mixed-signal design. An overview of our workflow can be found in Figure B.5. We will only use high level descriptions and a few code examples, we omit the fully automated scripts for the sake of brevity/readability. However, our workflow should be reproducible with a few weeks effort (we recommend consulting digital VLSI research groups if you want to implement this workflow at your home institution). A lot of this workflow, especially synthesis and place-and-route, can be referenced from Erik Brunvard's book "Digital

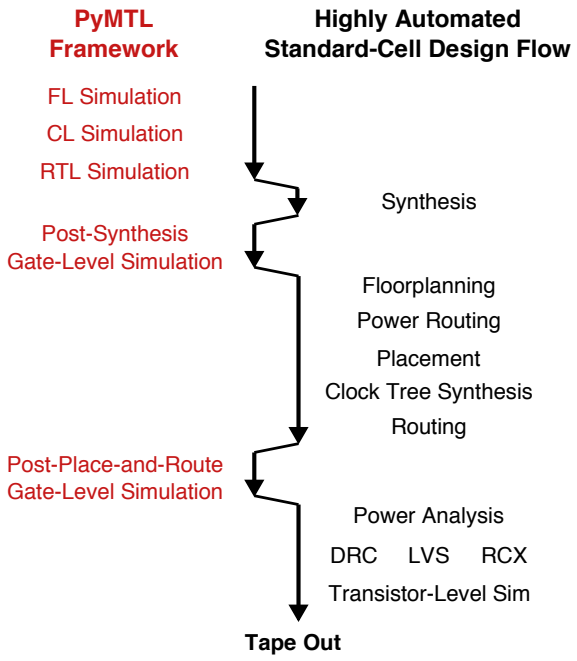


Figure B.1: Here is the high level overview of our design workflow. [Figure courtesy of Christopher Torng]

VLSI Chip Design with Cadence and Synopsys CAD Tools" [21].

B.2.1 Required Operating System Environment, Software Tools and File Formats

The computing environment and tools required to support this workflow is the most difficult part to maintain. We operate in a Linux-based environment with Cadence Virtuoso[2] for analog and mixed-signal circuit design, Synopsys Design Compiler[3] for synthesis, and

[2] https://www.cadence.com/content/cadence-www/global/en_US/home/tools/custom-ic-analog-rf-design.html

[3] http://www.synopsys.com/Tools/Implementation/RTLSynthesis/Pages/default.aspx

Synopys IC Compiler[4] for place-and-route. In addition, we utilize Cadence Encounter Library Characterization tool[5] for standard cell characterization, Calibre DRC and LVS from MentorGraphics[6] to verify our final chip layout. See [21] for a complete list of system dependencies.

For workflows with the ultimate aim to fabricate chips, the institution will need a Process Design Kit or PDK for a CMOS technology node. Due to copyright issues, we are not able to provide any information about these PDKs, but a PDK should come with its own standard cell library that is compatible with the Synopsys digital synthesis and place-and-route tools.

B.2.2 Characterizing Standard Cells

For creating digital blocks to perform computing such as greatest common divisors, FFT, sorting algorithms etc., its necessary to compose standard cells such as NOT, NAND, NOR, flip-flops, etc. into larger digital gates and modules. Normally, this is provided as a standard cell library with files Library Exchange Format (LEF) and LIB files. LEF describe the geometry of the physical layout of the cells, and are used during place-and-route. LIB describe the timings and simulation behavior of cells, and are used in synthesis.

If a standard cell library is not provided or additional standard cells are needed to be custom made, we need to generate our own LEF and LIB files. Following cell design and layout in Cadence Virtuoso (for an overview, consult [76]), we layout a digital circuit. We

[4] http://www.synopsys.com/Tools/Implementation/PhysicalImplementation/Pages/default.aspx

[5] https://www.cadence.com/content/cadence-www/global/en_US/home/tools/custom-ic-analog-rf-design/library-characterization.html

[6] https://www.mentor.com/products/ic_nanometer_design/verification-signoff/physical-verification/

.lib file

```
/* Characterization for a 3-input NAND gate */
cell ( NAND3X0 ) {

  /* Overall characterization */
  cell_footprint     : "nand3x0";
  area               : 7.3728;
  cell_leakage_power : 9.151417e+04;

  /* Characterization for input pin IN1 */
  pin ( IN1 ) {
    direction : "input";

    /* Fixed input capacitance */
    capacitance : 2.190745;

    /* Transient capacitance values */
    fall_capacitance : 2.212771;
    rise_capacitance : 2.168719;
```

ste'11]

Figure B.2: Sample portion of a LIB file describing timing characteristics for digital gae.

use a tool called *Abstract* to generate the LEF files directly in Virtuoso (see Section 10.4 of Brunvard [21] for detailed steps).

To generate LIB files, we utilize Cadence Encounter Library Characterization Tool (ELC) to generate LIB files. Note that you have to use existing SPICE or SPECTRE models that come with the PDK in generation of LIB files. See [21] for detailed scripts on how to perform this.

B.2.3 PyMTL to Verilog RTL

To design the digital blocks, we utilize the PyMTL hardware modeling framework [130] to generate synthesizable Verilog RTL. PyMTL enables faster research prototyping by al-

```
from pymtl import *
class DivUnit( Model ):
  def __init__( s, nbits ):
    # Port based interface
    s.dividend  = InPort ( Bits(nbits*2) )
    s.divisor   = InPort ( Bits( nbits ) )
    s.remainder = OutPort( Bits( nbits ) )
    s.quotient  = OutPort( Bits( nbits ) )
    # Establish input registers
    @s.tick_rtl
    def block1():
      s.r_in_reg.next = s.dividend;
      s.d_in_reg.next = s.divisor;
    # Intantiate partial quotients
    s.part_quots = [ partquot(nbits) for x in xrange(nbits+1) ]
```

Figure B.3: Sample pseudo-code of PyMTL for a fixed point divider.

lowing less low level code, a suite of testing tools including PyTest, and an active computer architecture research community that employs it (see https://github.com/cornell-brg/pymtl). PyMTL allows the description of digital circuitry at the functional level (FL), cycle level (CL), and register transfer level (RTL), multiple granularity levels that allows researchers to pick how detailed they want to simulate their designs at. See Figure B.3 for some pseudo-code of PyMTL for a 8 bit fixed point divider circuit. We note that building such a divider by custom design and hand layout would take several hours, while this method enables fully automated layout in seconds with interfaced testing to other digital blocks.

PyMTL code is then compiled to Verilog RTL that is used as the input to the Synopsys Design Compiler tool for design synthesis.

B.2.4 Synthesis and Place-and-Route

Digital synthesis is the process of translating RTL to Boolean logic which is then implemented using digital modules composed of standard cells. See [21] for a detailed description and sample scripts in order to do so. Our workflow leverages these scripts in order to push our RTL through to yield a gate level netlist. This gate level netlist is technology dependent, and can be used for simulation, power analysis, and is the primary input for place-and-route for chip layout and fabrication.

After a design is synthesized, it is fed through Synopsys IC compiler for automated place-and-route. See [21] for example scripts. Here is where we do power management and clock timing, especially important for large scale digital designs such as the microprocessor we detail in the case study. The output of IC compiler is a GDS file which can be sent to a foundtry to be fabricated in CMOS.

B.2.5 Interfacing with Mixed-Signal Design

After GDS output in the place-and-route stage, we then import GDS back into Cadence Virtuoso to interface with analog circuitry. We also perform Calibre DRC and LVS checks on this layout to make sure it will be compatible with the CMOS fabrication process.

Of course, an ideal workflow would require going through this entire process once. However, in practice, multiple iteration cycles are necessary to ensure the different stages are compatible with one another, in particular those that are technology and tool dependent. This is a limitation of the current workflow that we do not address.

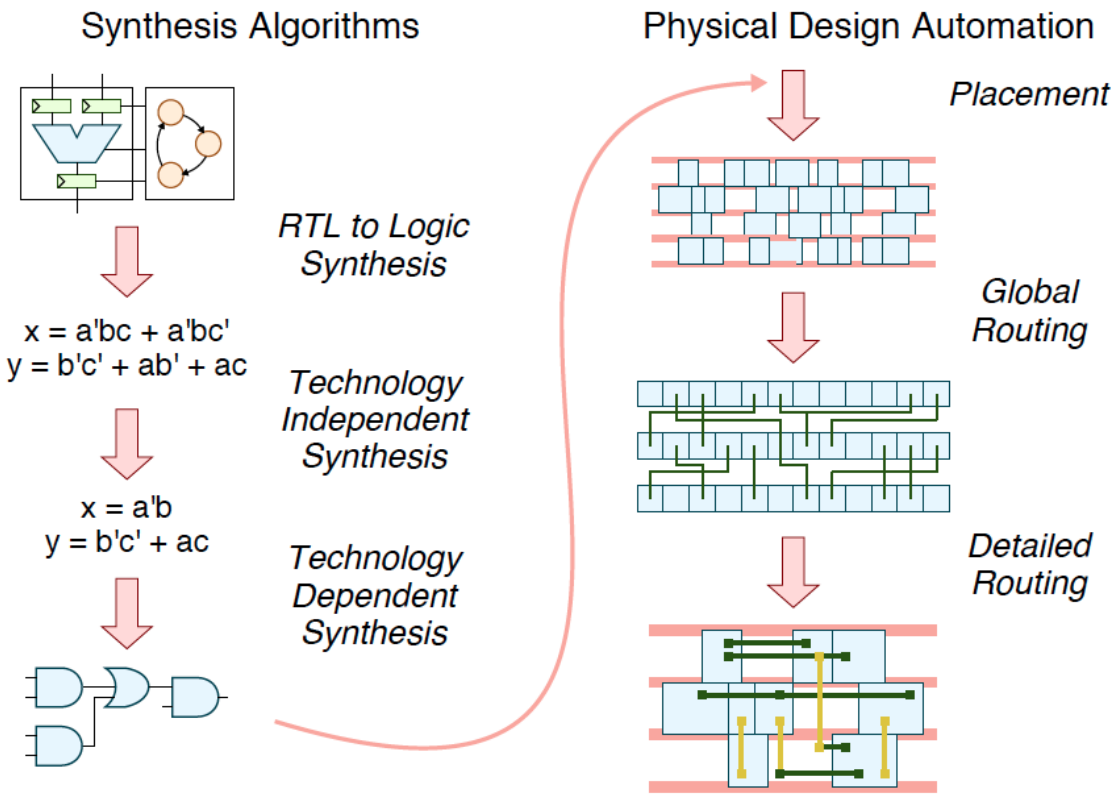


Figure B.4: A conceptual flowchart describing how RTL is passed through synthesis and place-and-route to yield physical layout that can be fabricated. [Figure courtesy of Christopher Batten]

B.3 Physical Validation of Design Flow

To validate our workflow, we study two examples: a microprocessor and a small test structure for Angle Sensitive Photogates for depth field imaging. We believe these cases show the versatility and complexity that the workflow can handle. Using a small team of less than 10 people (with only one student being experienced with actual tapeouts), we were able to tapeout these two examples in a 130nm BiCMOS process in a few months.

B.3.1 Processor

In Figure B.6, we show our fabricated 2x2 mm, 1.3M-transistor chip in IBM 130nm that was implemented using our design flow with PyMTL[7] The chip is a pipelined 32-bit RISC processor with a custom low voltage dynamic swing (LVDS) clock receiver, 16KB of on-chip SRAM, and a sorting accelerator generated using commercial C-to-RTL high-level synthesis tools. For more information about the design of the processor and its tapeout, see [35]. While this chip was not designed for computational imaging, we can imagine a future where on-board sensor processing can be performed using advanced computer microarchitectures. This synergy between image sensing and digital CMOS design can help pave the way for more interesting visual computing systems.

[7]This chip was developed in collaboration with Christopher Torng, Moyang Wang, Bharath Sudheendra, Nagaraj Murali, Shreesha Srinath, Taylor Pritchard, Robin Ying, and Christopher Batten.

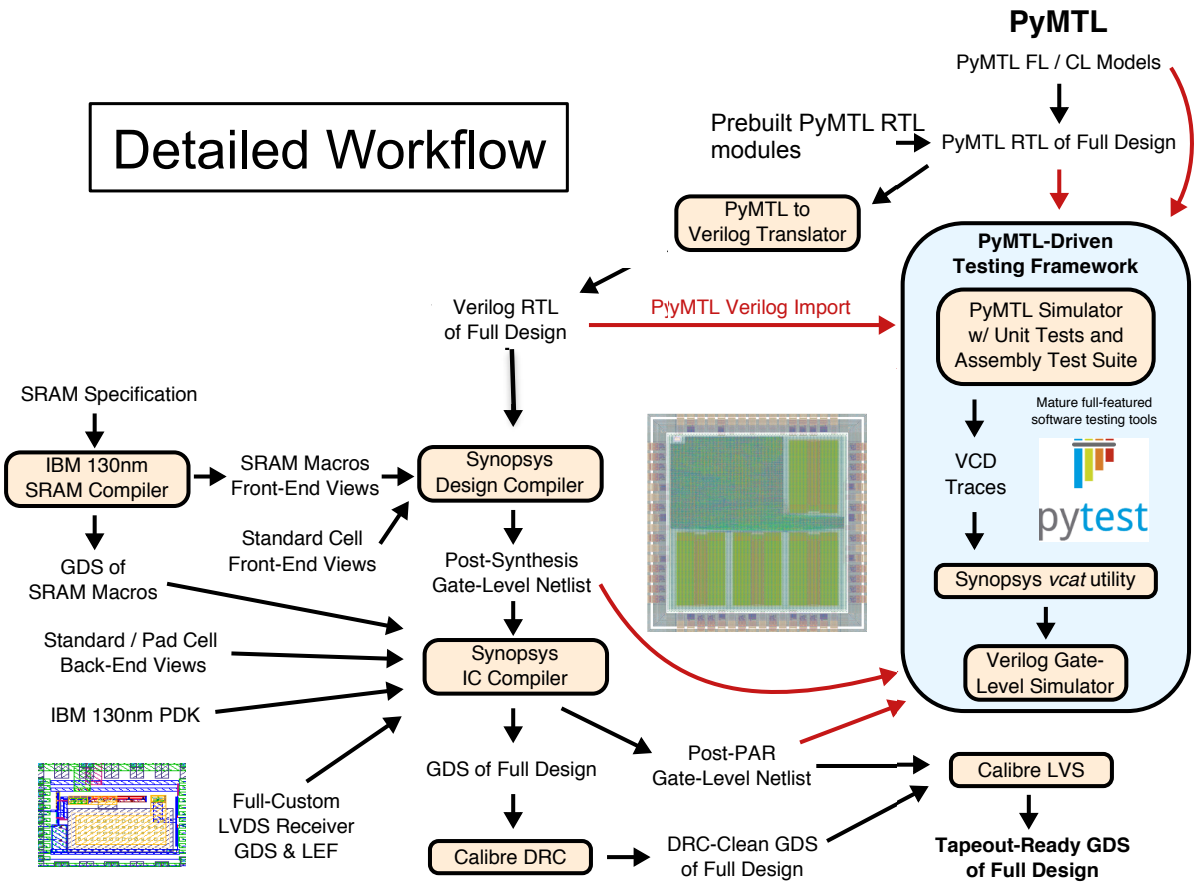


Figure B.5: The detailed process flow including on-chip SRAM integration, interfacing with custom analog circuit blocks (such as the LVDS receiver), and a testing framework for PyMTL. We omit steps related to FPGA emulation and HLS tools for advanced digital design, but refer to [35] for the full description. [Figure courtesy of Christopher Torng]

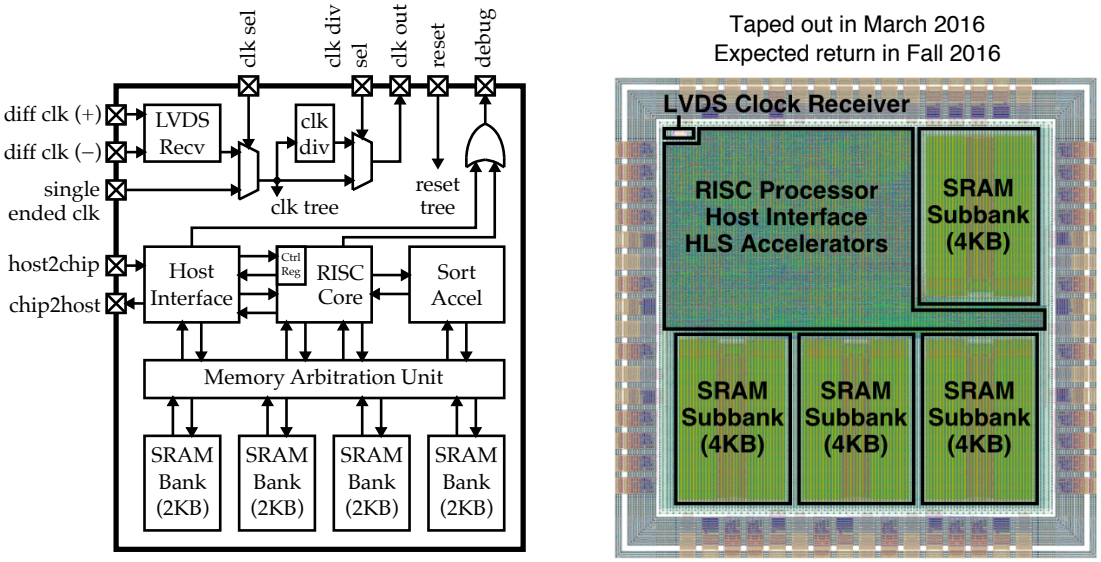


Figure B.6: Pipeline RISC processor using our design flow.

B.3.2 Test System for Depth Field Imaging

We also taped out a set of Angle Sensitive Photogates as described in Ch. 5. These pixels were designed in Cadence Virtuoso using analog design techniques. A custom amplifier, similar to the design presented in [9], was built that helps extend the dynamic range of ASPs by performing common mode resetting. This helps provide robustness against ambient light for ASP pixels in general. However, controlling this amplifier requires a series of digital signals as illustrated in the timing diagram of Figure B.7. We thus generate a control unit for this amplifier using the design flow, which saved several days of custom design work normally. We also implemented an 8 bit fixed point divider to perform quadrature division for ASP [196] using the design flow.

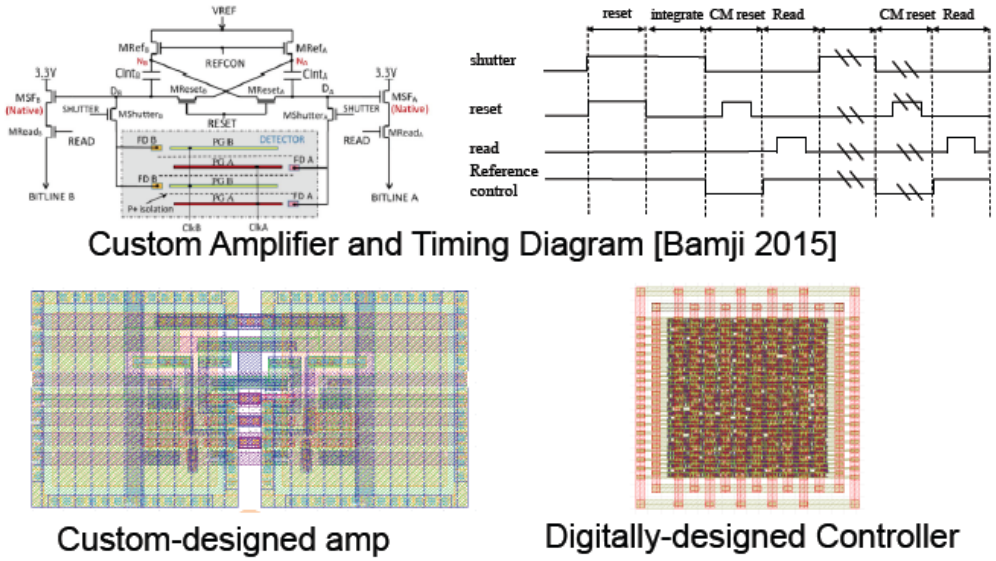


Figure B.7: Amplifier design and timing diagram [9] is shown, along with corresponding custom layout and digitally designed control unit using the workflow.

B.4 Future work

While this design flow does allow easier research prototyping and fabrication of mixed-signal chips, it does have several limitations that need to be addressed. The reliance on proprietary software tools for simulation limits open access and reproducibility in the research commmunity. It is our hope that future open source simulators, synthesis, and place-and-route tools using freely available PDKs will help allow researchers outside of VLSI be able to test out ideas in hardware.

For mixed-signal design, we still do not have adequate tools for simulating digital designed blocks interfacing with the corresponding analog circuits. This yields the potential for errors at these interfaces. Further, we still do not simplify the design of circuitry like ADCs and DACs that are critical for mixed-signal design, these are still the domain of

mixed-signal circuit experts.

However, we are encouraged by the development of this toolflow and the two case studies we taped out. Doing vertically integrated research is challenging, but we hope more interdisciplinary teams form to realize new exciting systems, particularly in the field of visual computing.

BIBLIOGRAPHY

[1] Edward H Adelson and James R Bergen. *The plenoptic function and the elements of early vision*.

[2] Edward H Adelson and John YA Wang. Single lens stereo with a plenoptic camera. *IEEE Trans. PAMI*, 14(2):99–106, 1992.

[3] Amit Agrawal, Ramesh Raskar, and Rama Chellappa. What is the range of surface reconstructions from a gradient field? In *European Conference on Computer Vision*, pages 578–591. Springer, 2006.

[4] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-svd: Design of dictionaries for sparse representation. *Proceedings of SPARS*, 5:9–12, 2005.

[5] Nicholas Antipa, Sylvia Necula, Ren Ng, and Laura Waller. Single-shot diffuser-encoded light field imaging. In *2016 IEEE International Conference on Computational Photography (ICCP)*, pages 1–11. IEEE, 2016.

[6] Amit Ashok and Mark A Neifeld. Compressive light field imaging. In *SPIE Defense, Security, and Sensing*, pages 76900Q–76900Q. International Society for Optics and Photonics, 2010.

[7] Gary A Atkinson and Edwin R Hancock. Recovery of surface orientation from diffuse polarization. *IEEE Transactions on image Processing*, 15(6):1653–1664, 2006.

[8] D Babacan, Reto Ansorge, Martin Luessi, Pablo Ruiz, Rafael Molina, and A Katsaggelos. Compressive light field sensing. 2012.

[9] Cyrus Bamji, Patrick O'Connor, Tamer Elkahtib, Swati Mehta, Barry Thompson, Lawrence Prather, Dane Snow, Onur Can Akkaya, Andy Daniel, Andrew D. Payne, Travis Perry, Mike Fenton, and Vei-Hang Chan. A 0.13um cmos system-on-chip for a 512x424 time-of-flight image sensor with multi-frequency photo-demodulation up to 130 mhz and 2 gs/s adc. *IEEE Journal of Solid-State Circuits*, 50:303–319, 2015.

[10] Richard G Baraniuk, Volkan Cevher, Marco F Duarte, and Chinmay Hegde. Model-based compressive sensing. *IEEE Transactions on Information Theory*, 56(4):1982–2001, 2010.

[11] Patrick Bardow, Andrew J. Davison, and Stefan Leutenegger. Simultaneous optical flow and intensity estimation from an event camera. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[12] Bryce E Bayer. Color imaging array, July 1976. US Patent 3,971,065.

[13] Brian C Becker and Enrique G Ortiz. Evaluating open-universe face identification on the web. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*, pages 904–911. IEEE, 2013.

[14] Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. *Unsupervised and Transfer Learning Challenges in Machine Learning*, 7:19, 2012.

[15] Ayush Bhandari, Achuta Kadambi, and Ramesh Raskar. Sparse linear operator identification without sparse regularization? applications to mixed pixel problem in time-of-flight/range imaging. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 365–369. IEEE, 2014.

[16] Ayush Bhandari, Achuta Kadambi, Refael Whyte, Christopher Barsi, Micha Feigin, Adrian Dorrington, and Ramesh Raskar. Resolving multipath interference in time-of-flight imaging via modulation frequency diversity and sparse regularization. *Optics Letters*, 39(6):1705–1708, 2014.

[17] Tom E Bishop, Sara Zanetti, and Paolo Favaro. Light field superresolution. In *Proc. ICCP*, pages 1–9. IEEE, 2009.

[18] Bernhard E Boser, Eduard Sackinger, Jane Bromley, Yann Le Cun, and Lawrence D Jackel. An analog neural network processor with programmable topology. *Solid-State Circuits, IEEE Journal of*, 26(12):2017–2025, 1991.

[19] S Boyd, N Parikh, E Chu, B Peleato, and J Eckstein. Matlab scripts for alternating direction method of multipliers. Technical report, Technical report, http://www. stanford. edu/boyd/papers/admm, 2012.

[20] Michael Broxton, Logan Grosenick, Samuel Yang, Noy Cohen, Aaron Andalman, Karl Deisseroth, and Marc Levoy. Wave optics theory and 3-d deconvolution for the light field microscope. *Optics express*, 21(21):25418–25439, 2013.

[21] Erik Brunvand. *Digital VLSI chip design with Cadence and Synopsys CAD tools*. Addison-Wesley, 2010.

[22] Harry E Burton. Euclids optics. *Journal of the Optical Society*, 35(5):357–72, 1945.

[23] E. Candès, J. Romberg, and T. Tao. Stable Signal Recovery from Incomplete and Inaccurate Measurements. *Comm. Pure Appl. Math.*, 59:1207–1223, 2006.

[24] Emmanuel J Candes. The restricted isometry property and its implications for compressed sensing. *Comptes Rendus Mathematique*, 346(9):589–592, 2008.

[25] Emmanuel J Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. 52(2):489–509, 2006.

[26] Emmanuel J Candes and Terence Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? 52(12):5406–5425, 2006.

[27] Emmanuel J Candès and Michael B Wakin. An introduction to compressive sampling. *IEEE signal processing magazine*, 25(2):21–30, 2008.

[28] Andrew S Cassidy, Paul Merolla, John V Arthur, Steve K Esser, Bryan Jackson, Rodrigo Alvarez-Icaza, Pallab Datta, Jun Sawada, Theodore M Wong, Vitaly Feldman, et al. Cognitive computing building block: A versatile and efficient digital neuron model for neurosynaptic cores. In *Neural Networks (IJCNN), The 2013 International Joint Conference on*, pages 1–10. IEEE, 2013.

[29] Sek M Chai, Antonio Gentile, Wilfredo E Lugo-Beauchamp, Javier Fonseca, Jose L Cruz-Rivera, and D Scott Wills. Focal-plane processing architectures for real-time hyperspectral image processing. *Applied Optics*, 39(5):835–849, 2000.

[30] Ayan Chakrabarti. Learning sensor multiplexing design through back-propagation. In *Advances in Neural Information Processing Systems*, 2016.

[31] Julie Chang, Isaac Kauvar, Xuemei Hu, and Gordon Wetzstein. Variable aperture light field photography: Overcoming the diffraction-limited spatio-angular resolution tradeoff. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3737–3745, 2016.

[32] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference*, 2014.

[33] Huaijin G. Chen, Suren Jayasuriya, Jiyue Yang, Judy Stephen, Sriram Sivaramakrishnan, Ashok Veeraraghavan, and Alyosha Molnar. Asp vision: Optically computing the first layer of convolutional neural networks using angle sensitive pixels. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[34] Wenlin Chen, James T. Wilson, Stephen Tyree, Kilian Q. Weinberger, and Yixin Chen. Compressing convolutional neural networks. *CoRR*, abs/1506.04449, 2015.

[35] Bharath Sudheendra Nagaraj Murali Suren Jayasuriya Shreesha Srinath Taylor Pritchard Robin Ying Christopher Torng, Moyang Wang and Christopher Batten. *Experiences Using a Novel Python-Based Hardware Modeling Framework for Computer Architecture Test Chips*. 28th ACM/IEEE Symposium on High Performance Chips (HOT CHIPS'16) Student Poster Session, 2016.

[36] Matthieu Courbariaux, Itay Hubara, COM Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks: Training neural networks with weights and activations constrained to+ 1 or-.

[37] S. Cova, M. Ghioni, A. Lacaita, C. Samori, and F. Zappa. Avalanche photodiodes and quenching circuits for single-photon detection. *Appl. Opt.*, 35(12):1956–1976, Apr 1996.

[38] Y. Le Cun, B. Boser, J. S. Denker, R. E. Howard, W. Habbard, L. D. Jackel, and D. Henderson. Advances in neural information processing systems 2. chapter Handwritten Digit Recognition with a Back-propagation Network, pages 396–404. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990.

[39] Dengxin Dai, Yujian Wang, Yuhua Chen, and Luc Van Gool. How useful is image super-resolution to other vision tasks? 2016.

[40] Frank Dellaert, Steven M. Seitz, Charles E. Thorpe, and Sebastian Thrun. Structure from motion without correspondence. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2000.

[41] David Donoho. Compressed Sensing. *IEEE Trans. Inform. Theory*, 52(4):1289–1306, 2006.

[42] David L Donoho, Arian Maleki, and Andrea Montanari. Message-passing algorithms for compressed sensing. *Proceedings of the National Academy of Sciences*, 106(45):18914–18919, 2009.

[43] Alexey Dosovitskiy, Jost Tobias Springenberg, and Thomas Brox. Learning to generate chairs with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1538–1546, 2015.

[44] David Droeschel, Dirk Holz, and Sven Behnke. Multi-frequency phase unwrapping for time-of-flight cameras. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1463–1469. IEEE, 2010.

[45] David Droeschel, Dirk Holz, and Sven Behnke. Probabilistic phase unwrapping for time-of-flight cameras. In *Proc. of Joint 41st Int. Symposium on Robotics and 6th German Conference on Robotics (ISR/ROBOTIK)*, pages 1–7. VDE, 2010.

[46] Frédo Durand, Nicolas Holzschuch, Cyril Soler, Eric Chan, and François X Sillion. A frequency analysis of light transport. In *ACM Trans. Graph. (SIGGRAPH)*, volume 24, pages 1115–1126, 2005.

[47] Abbas El Gamal and Helmy Eltoukhy. Cmos image sensors. *IEEE Circuits and Devices Magazine*, 21(3):6–20, 2005.

[48] Gabriele Fanelli, Matthias Dantone, Juergen Gall, Andrea Fossati, and Luc Van Gool. Random forests for real time 3d face analysis. *Int. J. Comput. Vision*, 101(3):437–458, February 2013.

[49] Clément Farabet, Cyril Poulet, and Yann LeCun. An fpga-based stream processor for embedded real-time vision with convolutional networks. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 878–885. IEEE, 2009.

[50] Nabil H Farhat, Demetri Psaltis, Aluizio Prata, and Eung Paek. Optical implementation of the hopfield model. *Applied Optics*, 24(10):1469–1475, 1985.

[51] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[52] David A Forsyth and Jean Ponce. *Computer vision: a modern approach*. Prentice Hall Professional Technical Reference, 2002.

[53] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376, 2010.

[54] Orazio Gallo, Iuri Frosio, Leonardo Gasparini, Kari Pulli, and Massimo Gottardi. Retrieving gray-level information from a binary sensor and its application to gesture detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 21–26, 2015.

[55] Leonardo Gasparini, Roberto Manduchi, and Massimo Gottardi. An ultra-low-power contrast-based integrated camera node and its application as a people counter. In *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, pages 547–554. IEEE, 2010.

[56] Leonardo Gasparini, Roberto Manduchi, Massimo Gottardi, and Dario Petri. An ultralow-power wireless camera node: Development and performance analysis. *IEEE Transactions on Instrumentation and Measurement*, 60(12):3824–3832, 2011.

[57] Nahum Gat, Gordon Scriven, John Garman, Ming De Li, and Jingyi Zhang. Development of four-dimensional imaging spectrometers (4d-is). In *SPIE Optics+ Photonics*, pages 63020M–63020M. International Society for Optics and Photonics, 2006.

[58] Elad Gilboa, John P Cunningham, Arye Nehorai, and Viktor Gruev. Image interpolation and denoising for division of focal plane sensors using gaussian processes. *Optics express*, 22(12):15277–15291, 2014.

[59] Patrick Robert Gill, Changhyuk Lee, Dhon-Gue Lee, Albert Wang, and Alyosha Molnar. A microscale camera using direct fourier-domain scene capture. *Optics letters*, 36(15):2949–2951, 2011.

[60] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):142–158, 2016.

[61] John Godbaz, Michael Cree, and Adrian Dorrington. Extending amcw lidar depth-of-field using a coded aperture. *Asian Conference on Computer Vision*, pages 397–409, 2010.

[62] S. Burak Gokturk, Hakan Yalcin, and Cyrus Bamji. A time-of-flight depth sensor - system description, issues and solutions. *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 3:35, 2004.

[63] Yunchao Gong, Liu Liu, Ming Yang, and Lubomir Bourdev. Compressing deep

convolutional networks using vector quantization. *arXiv preprint arXiv:1412.6115*, 2014.

[64] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[65] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *Proc. SIGGRAPH*, pages 43–54, 1996.

[66] M. Gottardi, N. Massari, and S.A. Jawed. A 100 μW 128$\times$64 pixels contrast-based asynchronous binary vision sensor for sensor networks applications. *IEEE Journal of Solid-State Circuits*, 44(5):1582–1592, 2009.

[67] Viktor Gruev and Ralph Etienne-Cummings. Implementation of steerable spatiotemporal image filters on the focal plane. *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, 49(4):233–244, 2002.

[68] Viktor Gruev, Ralph Etienne-Cummings, and Timmer Horiuchi. Linear current mode imager with low fix pattern noise. In *Circuits and Systems, 2004. ISCAS'04. Proceedings of the 2004 International Symposium on*, volume 4, pages IV–860. IEEE, 2004.

[69] Viktor Gruev, Alessandro Ortu, Nathan Lazarus, Jan Van der Spiegel, and Nader Engheta. Fabrication of a dual-tier thin film micropolarization array. *Optics express*, 15(8):4994–5007, 2007.

[70] Viktor Gruev, Rob Perkins, and Timothy York. Ccd polarization imaging sensor with aluminum nanowire optical filters. *Optics express*, 18(18):19087–19094, 2010.

[71] Seungyeop Han, Rajalakshmi Nandakumar, Matthai Philipose, Arvind Krishnamurthy, and David Wetherall. Glimpsedata: Towards continuous vision-based personal analytics. In *Proceedings of the 2014 workshop on physical analytics*, pages 31–36. ACM, 2014.

[72] Song Han, Xingyu Liu, Huizi Mao, Jing Pu, Ardavan Pedram, Mark A Horowitz, and William J Dally. Eie: efficient inference engine on compressed deep neural network. *arXiv preprint arXiv:1602.01528*, 2016.

[73] Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing

deep neural network with pruning, trained quantization and huffman coding. *CoRR*, abs/1510.00149, 2015.

[74] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[75] Paul Hasler. Low-power programmable signal processing. In *Fifth International Workshop on System-on-Chip for Real-Time Applications (IWSOC'05)*, pages 413–418. IEEE, 2005.

[76] Weste Neil HE et al. *Cmos Vlsi Design: A Circuits And Systems Perspective, 3/E*. Pearson Education India, 2006.

[77] Stefan Heber and Thomas Pock. Convolutional networks for shape from light field. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[78] E. Hecht. *Optics*. Addison-Wesley, 2002.

[79] Felix Heide, Matthias B Hullin, James Gregson, and Wolfgang Heidrich. Low-budget transient imaging using photonic mixer devices. *ACM Transactions on Graphics (TOG)*, 32(4):45, 2013.

[80] Felix Heide, Lei Xiao, Andreas Kolb, Matthias B Hullin, and Wolfgang Heidrich. Imaging in scattering media using correlation image sensors and sparse convolutional coding. *Optics Express*, 22(21):26338–26350, 2014.

[81] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox. Rgb-d mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments. *Int. J. Rob. Res.*, 31(5):647–663, April 2012.

[82] Michiel Hermans and Thomas Van Vaerenbergh. Towards trainable media: Using waves for neural network-style training. *arXiv preprint arXiv:1510.03776*, 2015.

[83] William F.J. Herrington. Micro-optic elements for a compact opto-electronic integrated neural coprocessor. *Thesis*.

[84] Matthew Hirsch, Sriram Sivaramakrishnan, Suren Jayasuriya, Albert Wang, Alyosha Molnar, Ramesh Raskar, and Gordon Wetzstein. A switchable light field camera architecture with angle sensitive pixels and dictionary-based sparse coding. In *Compu-*

tational Photography (ICCP), 2014 IEEE International Conference on, pages 1–10. IEEE, 2014.

[85] Benjamin Huhle, Timo Schairer, Philipp Jenke, and Wolfgang Straíer. Fusion of range and color images for denoising and resolution enhancement with a non-local filter. *Comput. Vis. Image Underst.*, 114(12):1336–1345, December 2010.

[86] Michael Iliadis, Leonidas Spinoulas, and Aggelos K Katsaggelos. Deep fully-connected networks for video compressive sensing. *arXiv preprint arXiv:1603.04930*, 2016.

[87] H Ives. Parallax Stereogram and Process of Making Same. US patent 725,567, 1903.

[88] Shahram Izadi, Andrew Davison, Andrew Fitzgibbon, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Dustin Freeman. Kinect Fusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera. *Proceedings of the 24th annual ACM symposium on User interface software and technology - UIST '11*, page 559, 2011.

[89] Wenzel Jakob. Mitsuba renderer, 2010.

[90] Suren Jayasuriya, Adithya Pediredla, Sriram Sivaramakrishnan, Alyosha Molnar, and Ashok Veeraraghavan. Depth fields: Extending light field techniques to time-of-flight imaging. In *3D Vision (3DV), 2015 International Conference on*, pages 1–9. IEEE, 2015.

[91] Suren Jayasuriya, Sriram Sivaramakrishnan, Ellen Chuang, Debashree Guruaribam, Albert Wang, and Alyosha Molnar. Dual light field and polarization imaging using cmos diffractive image sensors. *Optics Letters*, 40(10):2433–2436, 2015.

[92] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.

[93] Achuta Kadambi, Ayush Bhandari, Refael Whyte, Adrian Dorrington, and Ramesh Raskar. Demultiplexing illumination via low cost sensing and nanosecond coding. *IEEE International Conference on Computational Photography (ICCP)*, 2014.

[94] Achuta Kadambi, Vahe Taamazyan, Boxin Shi, and Ramesh Raskar. Polarized 3D:

High-Quality Depth Sensing with Polarization Cues. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[95] Achuta Kadambi, Refael Whyte, Ayush Bhandari, Lee Streeter, Christopher Barsi, Adrian Dorrington, and Ramesh Raskar. Coded time of flight cameras: sparse deconvolution to address multipath interference and recover time profiles. *ACM Trans. Graph.*, 32(6):167, 2013.

[96] Nima Khademi Kalantari, Ting-Chun Wang, and Ravi Ramamoorthi. Learning-based view synthesis for light field cameras. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2016)*, 35(6), 2016.

[97] M.H. Kamal, M. Golbabaee, and P. Vandergheynst. Light Field Compressive Sensing in Camera Arrays. In *Proc. ICASSP*, pages 5413 –5416, 2012.

[98] Shoji Kawahito, Izhal Abdul Halin, Takeo Ushinaga, Tomonari Sawada, Mitsuru Homma, and Yasunari Maeda. A cmos time-of-flight range image sensor with gates-on-field-oxide structure. *IEEE Sensors Journal*, 7(12):1578–1586, 2007.

[99] Changil Kim, Henning Zimmer, Yael Pritch, Alexander Sorkine-Hornung, and Markus Gross. Scene reconstruction from high spatio-angular resolution light fields. *ACM Trans. Graph.*, 32(4):73:1–73:12, July 2013.

[100] Hanme Kim, Ankur Handa, Ryad Benosman, Sio-Hoi Ieng, and Andrew J Davison. Simultaneous mosaicing and tracking with an event camera. *J. Solid State Circ*, 43:566–576, 2008.

[101] Hanme Kim, Stefan Leutenegger, and Andrew J Davison. Real-time 3d reconstruction and 6-dof tracking with an event camera. In *European Conference on Computer Vision*, pages 349–364. Springer, 2016.

[102] Yookyung Kim, Mariappan S Nadar, and Ali Bilgin. Compressed sensing using a Gaussian scale mixtures model in wavelet domain. pages 3365–3368. IEEE, 2010.

[103] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[104] Sanjeev J Koppal, Ioannis Gkioulekas, Terry Young, Hyunsung Park, Kenneth B Crozier, Geoffrey L Barrows, and Todd Zickler. Toward wide-angle microvision sensors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(12):2982–2996, 2013.

[105] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images, 2009.

[106] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[107] Kuldeep Kulkarni, Suhas Lohit, Pavan Turaga, Ronan Kerviche, and Amit Ashok. Reconnet: Non-iterative reconstruction of images from compressively sensed measurements. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[108] Kuldeep Kulkarni and Pavan Turaga. Reconstruction-free action inference from compressive imagers. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 38(4):772–784, 2016.

[109] Robert Lange. 3d time-of-flight distance measurement with custom solid-state image sensors in cmos/ccd-technology. *Diss., Department of Electrical Engineering and Computer Science, University of Siegen*, 2000.

[110] Robert Lange, Peter Seitz, Alice Biber, and Stefan Lauxtermann. Demodulation pixels in CCD and CMOS technologies for time-of-flight ranging. *Proceedings of SPIE*, 3965:177–188, 2000.

[111] Douglas Lanman, Ramesh Raskar, Amit Agrawal, and Gabriel Taubin. Shield fields: modeling and capturing 3d occluders. In *ACM Trans. Graph. (SIGGRAPH)*, volume 27, page 131, 2008.

[112] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[113] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[114] Yann LeCun, Corinna Cortes, and Christopher JC Burges. The mnist database of handwritten digits, 1998.

[115] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic sin-

gle image super-resolution using a generative adversarial network. *arXiv preprint arXiv:1609.04802*, 2016.

[116] Changhyuk Lee, Ben Johnson, and Alyosha Molnar. An on-chip 72x60 angle-sensitive single photon image sensor array for lens-less time-resolved 3-d fluorescence lifetime imaging. In *Symposium on VLSI Circuits*. IEEE, 2014.

[117] Walter D Leon-Salas, Sina Balkir, Khalid Sayood, Nathan Schemm, and Michael W Hoffman. A cmos imager with focal plane compression using predictive coding. *IEEE Journal of Solid-State Circuits*, 42(11):2555–2572, 2007.

[118] Anat Levin and Fredo Durand. Linear view synthesis using a dimensionality gap light field prior. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1831–1838. IEEE, 2010.

[119] Anat Levin, Rob Fergus, Frédo Durand, and William T. Freeman. Image and depth from a conventional camera with a coded aperture. *ACM Trans. Graph.*, 26(3), July 2007.

[120] Marc Levoy, Billy Chen, Vaibhav Vaish, Mark Horowitz, Ian McDowall, and Mark Bolas. Synthetic aperture confocal imaging. *ACM Trans. Graph.*, 23(3):825–834, August 2004.

[121] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proc. SIGGRAPH*, pages 31–42, 1996.

[122] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128$\times$ 128 120 db 15 μs latency asynchronous temporal contrast vision sensor. *Solid-State Circuits, IEEE Journal of*, 43(2):566–576, 2008.

[123] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128$\times$ 128 120 db 15 μs latency asynchronous temporal contrast vision sensor. *IEEE journal of solid-state circuits*, 43(2):566–576, 2008.

[124] Robert LiKamWa, Yunhui Hou, Yuan Gao, Mia Polansky, and Lin Zhong. Redeye: Analog convnet image sensor architecture for continuous mobile vision. In *Proceedings of ACM/IEEE Int. Symp. Computer Architecture (ISCA)*, 2016.

[125] Robert LiKamWa, Bodhi Priyantha, Matthai Philipose, Lin Zhong, and Paramvir Bahl. Energy characterization and optimization of image sensing toward continuous

mobile vision. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, pages 69–82. ACM, 2013.

[126] Robert LiKamWa, Zhen Wang, Aaron Carroll, Felix Xiaozhu Lin, and Lin Zhong. Draining our glass: An energy and heat characterization of google glass. In *Proceedings of 5th Asia-Pacific Workshop on Systems*. ACM, 2014.

[127] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *CoRR*, abs/1312.4400, 2013.

[128] Zhouhan Lin, Matthieu Courbariaux, Roland Memisevic, and Yoshua Bengio. Neural networks with few multiplications. *CoRR*, abs/1510.03009, 2015.

[129] Gabriel Lippmann. La Photographie Intégrale. *Academie des Sciences*, 146:446–451, 1908.

[130] Derek Lockhart, Gary Zibrat, and Christopher Batten. Pymtl: A unified framework for vertically integrated computer architecture research. In *2014 47th Annual IEEE/ACM International Symposium on Microarchitecture*, pages 280–292. IEEE, 2014.

[131] Suhas Lohit, Kuldeep Kulkarni, Pavan Turaga, Jian Wang, and Aswin Sankaranarayanan. Reconstruction-free inference on compressive measurements. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 16–24, 2015.

[132] A. Lumsdaine and T. Georgiev. The focused plenoptic camera. In *Proc. ICCP*, pages 1–8, 2009.

[133] Xiao-Jiao Mao, Chunhua Shen, and Yu-Bin Yang. Image restoration using convolutional auto-encoders with symmetric skip connections. *arXiv preprint arXiv:1606.08921*, 2016.

[134] Stephen Robert Marschner. *Inverse rendering for computer graphics*. PhD thesis, Cornell University, 1998.

[135] Kshitij Marwah, Gordon Wetzstein, Yosuke Bando, and Ramesh Raskar. Compressive light field photography using overcomplete dictionaries and optimized projections. *ACM Trans. Graph. (TOG)*, 32(4):46, 2013.

[136] Yoshitaka Miyatani, Souptik Barua, and Ashok Veeraraghavan. Direct face detection and video reconstruction from event cameras. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*. IEEE, 2016.

[137] Pavlo Molchanov, Xiaodong Yang, Shalini Gupta, Kihwan Kim, Stephen Tyree, and Jan Kautz. Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4207–4215, 2016.

[138] Ali Mousavi, Ankit B. Patel, and Richard G. Baraniuk. A deep learning approach to structured signal recovery. *CoRR*, abs/1508.04065, 2015.

[139] Shree K Nayar, Xi-Sheng Fang, and Terrance Boult. Separation of reflection components using color and polarization. *International Journal of Computer Vision*, 21(3):163–186, 1997.

[140] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[141] Ren Ng, Marc Levoy, Mathieu Brédif, Gene Duval, Mark Horowitz, and Pat Hanrahan. Light field photography with a hand-held plenoptic camera. *Computer Science Technical Report CSTR*, 2(11), 2005.

[142] Alireza Nilchi, Joseph Aziz, and Roman Genov. Focal-plane algorithmically-multiplying cmos computational image sensor. *Solid-State Circuits, IEEE Journal of*, 44(6):1829–1839, 2009.

[143] Peter O'Connor, Daniel Neil, Shih-Chii Liu, Tobi Delbruck, and Michael Pfeiffer. Real-time classification and sensor fusion with a spiking deep belief network. *Neuromorphic Engineering Systems and Applications*, page 61, 2015.

[144] United States. National Bureau of Standards and Fred Edwin Nicodemus. *Geometrical considerations and nomenclature for reflectance*, volume 160. US Department of Commerce, National Bureau of Standards, 1977.

[145] Bruno A Olshausen and David J Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

[146] Matthew O'Toole. *Optical Linear Algebra for Computational Light Transport*. PhD thesis, University of Toronto, 2016.

[147] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345–1359, 2010.

[148] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. *CVPR*, 2016.

[149] Andrew D Payne, Adrian PP Jongenelen, Adrian A Dorrington, Michael J Cree, and Dale A Carnegie. Multiple frequency range imaging to remove measurement ambiguity. *Proceedings of the 9th Conference on Optical 3-D Measurement Techniques*, 2009.

[150] Christian Perwass and Lennart Wietzke. Single Lens 3D-Camera with Extended Depth-of-Field. In *Proc. SPIE 8291*, pages 29–36, 2012.

[151] Phi-Hung Pham, Darko Jelaca, Clement Farabet, Berin Martini, Yann LeCun, and Eugenio Culurciello. Neuflow: Dataflow vision processing system-on-a-chip. In *Circuits and Systems (MWSCAS), 2012 IEEE 55th International Midwest Symposium on*, pages 1044–1047. IEEE, 2012.

[152] Demetri Psaltis, David Brady, Xiang-Guang Gu, and Steven Lin. Holography in artificial neural networks. *Nature*, 343(6256):325–330, 1990.

[153] Jonathan Millard Ragan-Kelley. *Decoupling algorithms from the organization of computation for high performance image processing*. PhD thesis, Ch. 2, pages 19–24, Massachusetts Institute of Technology, 2014.

[154] Rajeev Ramanath, Wesley E Snyder, Youngjun Yoo, and Mark S Drew. Color image processing pipeline. *IEEE Signal Processing Magazine*, 22(1):34–43, 2005.

[155] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[156] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[157] Mukul Sarkar, David San Segundo San Segundo Bello, Chris Van Hoof, and Albert Theuwissen. Integrated polarization analyzing cmos image sensor for material classification. *IEEE Sensors Journal*, 11(8):1692–1703, 2011.

[158] Yoav Y Schechner and Nir Karpel. Recovery of underwater visibility and structure by polarization analysis. *IEEE Journal of Oceanic Engineering*, 30(3):570–587, 2005.

[159] Yoav Y Schechner, Srinivasa G Narasimhan, and Shree K Nayar. Instant dehazing of images using polarization. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–325. IEEE, 2001.

[160] Yoav Y Schechner, Srinivasa G Narasimhan, and Shree K Nayar. Polarization-based vision through haze. *Applied optics*, 42(3):511–525, 2003.

[161] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. *CoRR*, abs/1503.03832, 2015.

[162] Rudolf Schwarte. New electro-optical mixing and correlating sensor: facilities and applications of the photonic mixer device (PMD). *Proceedings of SPIE*, 3100:245–253, 1997.

[163] Ana Serrano, Felix Heide, Diego Gutierrez, Gordon Wetzstein, and Belen Masia. Convolutional sparse coding for high dynamic range imaging. *Computer Graphics Forum*, 35(2), 2016.

[164] Premchandra M. Shankar, William C. Hasenplaugh, Rick L. Morrison, Ronald A. Stack, and Mark A. Neifeld. Multiaperture imaging. *Appl. Opt.*, 45(13):2871–2883, 2006.

[165] Jie Shen, Stefanos Zafeiriou, Grigoris G Chrysos, Jean Kossaifi, Georgios Tzimiropoulos, and Maja Pantic. The first facial landmark tracking in-the-wild challenge: Benchmark and results. In *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, pages 1003–1011. IEEE, 2015.

[166] Lixin Shi, Haitham Hassanieh, Abe Davis, Dina Katabi, and Fredo Durand. Light field reconstruction using sparsity in the continuous fourier domain. *ACM Transactions on Graphics (TOG)*, 34(1):12, 2014.

[167] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[168] Sriram Sivaramakrishnan, Albert Wang, Patrick Gill, and Alyosha Molnar. Design

and characterization of enhanced angle sensitive pixels. *Transactions on Electron Devices*, 63(1), 2016.

[169] Sriram Sivaramakrishnan, Albert Wang, Patrick R Gill, and Alyosha Molnar. Enhanced angle sensitive pixels for light field imaging. In *Proc. IEEE IEDM*, pages 8–6, 2011.

[170] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: Exploring photo collections in 3d. *ACM Trans. Graph.*, 25(3):835–846, July 2006.

[171] Shuochen Su, Felix Heide, Robin Swanson, Jonathan Klein, Clara Callenberg, Matthias Hullin, and Wolfgang Heidrich. Material classification using raw time-of-flight measurements. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[172] Atsushi Suzuki, Nobutaka Shimamura, Toshiki Kainuma, Naoki Kawazu, Chihiro Okada, Takumi Oka, Kensuke Koiso, Atsushi Masagaki, Yoichi Yagasaki, Shigeru Gonoi, et al. A 1/1.7-inch 20mpixel back-illuminated stacked cmos image sensor for new imaging applications. In *Solid-State Circuits Conference-(ISSCC), 2015 IEEE International*, pages 1–3. IEEE, 2015.

[173] Mate Szarvas, Akira Yoshizawa, Munetaka Yamamoto, and Jun Ogata. Pedestrian detection with convolutional neural networks. In *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, pages 224–229. IEEE, 2005.

[174] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014.

[175] Ryuichi Tadano, Adithya Pediredla, and Ashok Veeraraghavan. Depth selective camera: A direct, on-chip, programmable technique for depth selectivity in photography. In *IEEE International Conference on Computer Vision (ICCV)*, (Accepted) 2015.

[176] Akira Takahashi, Mitsunori Nishizawa, Yoshinori Inagaki, Musubu Koishi, and Katsuyuki Kinoshita. New femtosecond streak camera with temporal resolution of 180 fs, 1994.

[177] Jun Tanida, Tomoya Kumagai, Kenji Yamada, Shigehiro Miyatake, Kouichi Ishida, Takashi Morimoto, Noriyuki Kondou, Daisuke Miyazaki, and Yoshiki Ichioka. Thin observation module by bound optics (tombo): Concept and experimental verification. *Appl. Opt.*, 40(11):1806–1813, 2001.

[178] Michael W Tao, Sunil Hadap, Jitendra Malik, and Ravi Ramamoorthi. Depth from combining defocus and correspondence using light-field cameras. In *IEEE International Conference on Computer Vision (ICCV)*, pages 673–680. IEEE, 2013.

[179] Michael Wish Tao, Ravi Ramamoorthi, Jitendra Malik, and Alexei Alyosha Efros. *Unified Multi-Cue Depth Estimation from Light-Field Images: Correspondence, Defocus, Shading, and Specularity*. 2015.

[180] Sebastian Thrun. Exploring artificial intelligence in the new millennium. chapter Robotic Mapping: A Survey, pages 1–35. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003.

[181] Changpeng Ti, Ruigang Yang, James Davis, and Zhigeng Pan. Simultaneous time-of-flight sensing and photometric stereo with a single tof sensor. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4334–4342, 2015.

[182] Jack Tumblin, Amit Agrawal, and Ramesh Raskar. Why i want a gradient camera. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 103–110. IEEE, 2005.

[183] Shinji Umeyama and Guy Godin. Separation of diffuse and specular components of surface reflection by use of polarization and statistical analysis of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):639–647, 2004.

[184] Vaibhav Vaish, Marc Levoy, Richard Szeliski, C Lawrence Zitnick, and Sing Bing Kang. Reconstructing occluded surfaces using synthetic apertures: Stereo, focus and robust measures. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 2331–2338. IEEE, 2006.

[185] A. Vedaldi and K. Lenc. Matconvnet – convolutional neural networks for matlab. 2015.

[186] Ashok Veeraraghavan, Ramesh Raskar, Amit Agrawal, Ankit Mohan, and Jack Tumblin. Dappled photography: Mask enhanced cameras for heterodyned light fields and coded aperture refocusing. *ACM Trans. Graph. (SIGGRAPH)*, 26(3):69, 2007.

[187] Andreas Velten, Di Wu, Adrian Jarabo, Belen Masia, Christopher Barsi, Chinmaya Joshi, Everett Lawson, Moungi Bawendi, Diego Gutierrez, and Ramesh Raskar. Femto-photography: Capturing and visualizing the propagation of light. *ACM Transactions on Graphics (TOG)*, 32(4):44, 2013.

[188] Kartik Venkataraman, Dan Lelescu, Jacques Duparré, Andrew McMahon, Gabriel Molina, Priyam Chatterjee, Robert Mullis, and Shree Nayar. Picam: an ultra-thin high performance monolithic camera array. *ACM Trans. Graph. (SIGGRAPH Asia)*, 32(6):166, 2013.

[189] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408, 2010.

[190] A. Wang, S. Sivaramakrishnan, and A. Molnar. A 180nm cmos image sensor with on-chip optoelectronic image compression. In *Proc. IEEE Custom Integrated Circuits Conference (CICC)*, pages 1–4, 2012.

[191] Albert Wang, Patrick Gill, and Alyosha Molnar. Angle sensitive pixels in cmos for lensless 3d imaging. In *2009 IEEE Custom Integrated Circuits Conference*, pages 371–374. IEEE, 2009.

[192] Albert Wang, Patrick Gill, and Alyosha Molnar. Light field image sensors based on the talbot effect. *Applied optics*, 48(31):5897–5905, 2009.

[193] Albert Wang, Patrick R Gill, and Alyosha Molnar. An angle-sensitive cmos imager for single-sensor 3d photography. In *Proc. IEEE Solid-State Circuits Conference (ISSCC)*, pages 412–414. IEEE, 2011.

[194] Albert Wang, Sheila S Hemami, and Alyosha Molnar. Angle-sensitive pixels: a new paradigm for low-power, low-cost 2d and 3d sensing. In *IS&T/SPIE Electronic Imaging*, pages 828805–828805. International Society for Optics and Photonics, 2012.

[195] Albert Wang and Alyosha Molnar. Phase-based 3d optical flow sensors for motion detection. In *Sensors, 2011 IEEE*, pages 683–686. IEEE, 2011.

[196] Albert Wang and Alyosha Molnar. A light-field image sensor in 180 nm cmos. *Solid-State Circuits, IEEE Journal of*, 47(1):257–271, 2012.

[197] Ting-Chun Wang, Jun-Yan Zhu, Ebi Hiroaki, Manmohan Chandraker, Alexei Efros, and Ravi Ramamoorthi. A 4D light-field dataset and CNN architectures for material recognition. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2016.

[198] S. Wanner and B. Goldluecke. Variational light field analysis for disparity estimation and super-resolution. *IEEE Trans. PAMI*, 2013.

[199] David Weikersdorfer, David B Adrian, Daniel Cremers, and Jörg Conradt. Event-based 3d slam with a depth-augmented dynamic vision sensor. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 359–364. IEEE, 2014.

[200] David Weikersdorfer, Raoul Hoffmann, and Jörg Conradt. Simultaneous localization and mapping for event-based vision systems. In *International Conference on Computer Vision Systems*, pages 133–142. Springer, 2013.

[201] Alexander Wender, Julian Iseringhausen, Bastian Goldlücke, Martin Fuchs, and Matthias B. Hullin. Light field imaging through household optics. In David Bommes, Tobias Ritschel, and Thomas Schultz, editors, *Vision, Modeling & Visualization*, pages 159–166. The Eurographics Association, 2015.

[202] G. Wetzstein, I. Ihrke, and W. Heidrich. On Plenoptic Multiplexing and Reconstruction. *IJCV*, 101:384–400, 2013.

[203] Gordon Wetzstein. Synthetic light field archive. http://web.media.mit.edu/~gordonw/SyntheticLightFields/,.

[204] Bennett Wilburn, Neel Joshi, Vaibhav Vaish, Eino-Ville Talvala, Emilio Antunez, Adam Barth, Andrew Adams, Mark Horowitz, and Marc Levoy. High performance imaging using large camera arrays. *ACM Trans. Graph. (SIGGRAPH)*, 24(3):765–776, 2005.

[205] Lawrence B. Wolff. Polarization-based material classification from specular reflection. *IEEE transactions on pattern analysis and machine intelligence*, 12(11):1059–1071, 1990.

[206] Lawrence B Wolff and Todd A Mancini. Liquid-crystal polarization camera. In *Applications in Optical Science and Engineering*, pages 102–113. International Society for Optics and Photonics, 1992.

[207] Di Wu, Andreas Velten, Matthew O'toole, Belen Masia, Amit Agrawal, Qionghai Dai, and Ramesh Raskar. Decomposing global light transport using time of flight imaging. *Int. J. Comput. Vision*, 107(2):123–138, April 2014.

[208] G. Wyszecki and W.S. Stiles. *Color science: concepts and methods, quantitative data and formulae*. Wiley classics library. Wiley, 1982.

[209] Lei Xiao, Felix Heide, Matthew O'Toole, Andreas Kolb, Matthias B Hullin, Kyros Kutulakos, and Wolfgang Heidrich. Defocus deblurring and superresolution for time-of-flight depth cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2376–2384, 2015.

[210] Zhimin Xu and Edmund Y Lam. A high-resolution lightfield camera with dual-mask design. In *SPIE Optical Engineering+Applications*, pages 85000U–85000U, 2012.

[211] Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[212] Tao Yang, Wenguang Ma, Sibing Wang, Jing Li, Jingyi Yu, and Yanning Zhang. Kinect based real-time synthetic aperture imaging through occlusion. *Multimedia Tools and Applications*, pages 1–19, 2015.

[213] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *Acm computing surveys (CSUR)*, 38(4):13, 2006.

[214] Youngjin Yoon, Hae-Gon Jeon, Donggeun Yoo, Joon-Young Lee, and In So Kweon. Learning a deep convolutional network for light-field image super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 24–32, 2015.

[215] Timothy York, Samuel B Powell, Shengkui Gao, Lindsey Kahan, Tauseef Charanya, Debajit Saha, Nicholas W Roberts, Thomas W Cronin, Justin Marshall, Samuel Achilefu, et al. Bioinspired polarization imaging sensors: from circuits and optics to signal processing algorithms and biomedical applications. *Proceedings of the IEEE*, 102(10):1450–1469, 2014.

[216] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*, pages 3320–3328, 2014.

[217] Kaan Yücer, Alexander Sorkine-Hornung, Oliver Wang, and Olga Sorkine-Hornung. Efficient 3d object segmentation from densely sampled light fields with applications to 3d reconstruction. *ACM Trans. Graph.*, 35(3):22:1–22:15, March 2016.

[218] Zhengyun Zhang and Marc Levoy. Wigner distributions and how they relate to the light field. In *Computational Photography (ICCP), 2009 IEEE International Conference on*, pages 1–10. IEEE, 2009.

[219] Jiejie Zhu, Liang Wang, Ruigang Yang, and James Davis. Fusion of time-of-flight depth and stereo for high accuracy depth maps. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2008.

[220] Assaf Zomet and Shree K Nayar. Lensless imaging with a controllable aperture. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 339–346. IEEE, 2006.